

RESOURCE ALLOCATION FOR REGIONAL HURRICANE MITIGATION  
PLANNING

A Dissertation

Presented to the Faculty of the Graduate School  
of Cornell University

In Partial Fulfillment of the Requirements for the Degree of  
Doctor of Philosophy

by

Meredith Robin Legg

May 2011

© 2011 Meredith Robin Legg

# RESOURCE ALLOCATION FOR REGIONAL HURRICANE MITIGATION PLANNING

Meredith Legg, Ph. D.

Cornell University 2011

This dissertation introduces a linear program to help guide optimal expenditure of government spending for regional hurricane risk management and to provide insight into some of the complexities involved in designing and prioritizing regional mitigation policies and programs. Specifically, it aims to help answer the questions: (1) How much should be spent on mitigation and acquisition?; (2) What will the return on that investment be?; and (3) How should mitigation funds be spent (i.e., which buildings should be mitigated, how, and when)? The model considers damage from both high winds and storm surge flooding; includes a detailed assessment of the actual risk using a carefully selected set of hurricane scenarios to represent the regional hazard and a component-based damage model; and considers physically realistic mitigation strategies. A heuristic algorithm was developed to solve it for real, regional applications. A case study for residential woodframe buildings in Eastern North Carolina is presented. The case study suggests that spending on pre- and post-event mitigation and acquisition are all cost-beneficial to a point; if funds are spent systematically, much of the benefit can be obtained with a relatively small investment; and in just 30 years, the investment can reduce annual expected reconstruction expenditures substantially, a benefit that would continue into the future. The case study also suggests spending on a range of mitigation strategy types; that it is best to spend mitigation funds as early as possible, where the hazard is highest (i.e. along the

coast line); and that strategies affecting combinations of building components can be most cost-beneficial.

## BIOGRAPHICAL SKETCH

Meredith Legg attended Iowa State University and received a Bachelors of Science Degree in Civil and Environmental Engineering, emphasis in structural engineering, with distinction in December, 2004. She completed her M.S. in Civil Infrastructure Systems at Cornell University in August, 2007, and her PhD in Civil Infrastructure Systems at Cornell University in May, 2011.

I dedicate this dissertation to my beloved husband, Ryan Legg, who has been my  
unwaivering support through three graduations.

## ACKNOWLEDGMENTS

The authors thank the National Science Foundation (CMMI-0555738) for financial support of this research. Thank you to Brian Blanton for his help developing the wind and surge hazard data. Thank you to my advisors, Linda Nozick, Rachel Davidson, Oliver Gao and Thomas O'Rourke. Thank you to Palm Apivatanagul for the use of the hazard events from his M.S. thesis. This support is gratefully acknowledged, but I take sole responsibility for the content of the paper.

## TABLE OF CONTENTS

Biographical Sketch	iii
Dedication	iv
Acknowledgements	v
Table of Contents	vi
List of Figures	ix
List of Tables	xii
Chapter 1 Introduction	1
Chapter 2 The Challenge of Regional Hurricane Risk Management	4
Chapter 3 Resource Allocation for Natural Disaster Risk Management	6
3.1 Previously Developed Models	6
3.2 Comparison with the New Model	8
Chapter 4 Optimization Model	10
4.1 Definitions and Scope	10
4.2 Modeling Approach	13
4.3 Model Formulation	14
4.4 Solution Procedure	19
4.4.1 Benefit-Cost Metric Definitions	19
4.4.2 Heuristic Algorithm	22
4.4.3 Comparison Between Heuristic and Linear Program Solver	27
Chapter 5 Inputs	28
5.1 Hurricane Scenarios	28
5.2 Building Categories, Components, and Possible Component Configurations	31
5.3 Building Resistances and Mitigation Alternatives	34



5.4 Initial Building Inventory	36
5.5 Component and Building Damage States	38
5.6 Damage Estimates	40
5.6.1 Wind Damage Model	41
5.6.2 Flood Damage Estimation	46
5.7 Mitigation Costs, Acquisition Costs, Reconstruction Costs, and Budgets	46
Chapter 6 Case Study	49
6.1 Scope and Overview	49
6.2 How Much to Spend	51
6.3 What you get for the Investment	53
6.4 How to Spend It	57
6.4.1 By Type of Investment and Targeted Hazard	58
6.4.2 By Timing of Investment	60
6.4.3 By Geographic Location	63
6.4.4 By Specific Mitigation Strategy	64
Chapter 7 Conclusions and Future Work	70
7.1 Conclusions	70
7.2 Future Work	72
Appendix A Model Buildings	75
Appendix B Building Resistances and Mitigation Alternatives	78
Appendix C Building Damage State Definitions	88
Appendix D Wind Load Calculations	106
Appendix E Wind Damage Model Calibration	119
Appendix F Flood Fragility Curves	127
Appendix G Cost Estimates	133

Appendix H Loss Model Matlab Code	149
Appendix I Loss Model Matlab User Manual	211
Appendix J Resource Allocation Heuristic Matlab Code	216
Appendix K Resource Allocation Heuristic User Manual	237
References	238

## LIST OF FIGURES

Figure 1	Evolution of building inventory in time period $t$	13
Figure 2	Heuristic solution algorithm	23
Figure 3	Summary of the 68 building resistances $c$ considered in the case study	35
Figure 4	Summary of the 124 mitigation alternatives considered in the case study	37
Figure 5	Schematic of wind damage estimation model	42
Figure 6	Eight building orientations relative to the wind direction	43
Figure 7	Eastern North Carolina study area with census tract outlines	49
Figure 8	Recommended expenditures vs. annual pre- and post-event budgets, assuming pre- and post-event budgets are equal.	53
Figure 9	Overall benefit-cost ratio and total (mitigation, acquisition, and reconstruction) expenditures vs. annual pre- and post-event budgets, assuming pre- and post-event budgets are equal.	55
Figure 10	Expected Annual Reconstruction Expenditures for specific mitigation and acquisition annual budgets	57
Figure 11	Distribution of total expenditures during the 30-year time horizon in terms of (a) dollars and (b) number of buildings affected.	59
Figure 12	Recommended pre-event spending by year and type of mitigation, for base case analysis with \$8 million pre- and post-event annual budgets	62
Figure 13	Recommended post-event spending by year and type of mitigation, for base case analysis with \$8 million pre- and post-event annual budgets	63
Figure 14	Map of study area indicating census tracts where pre-event mitigation spending is recommended.	64

Figure 15	Percentage of buildings mitigated by the mitigation strategy used, for pre-event mitigation and post-event mitigation (RTW refers to the roof-to-wall connection component)	67
Figure A.1	Model building dimensions	76
Figure A.2	Model building opening locations	76
Figure B.1	Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study	84
Figure B.2	Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study	85
Figure B.3	Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study	86
Figure B.4	Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study	87
Figure D.1	Walls based on the wind direction corresponding to the gust effect factor and external pressure coefficients	110
Figure D.2	Building surfaces corresponding to external pressure coefficients based on the wind direction	113
Figure D.3	Diagrams corresponding to the calculation of shear wall loads	114
Figure E.1	Comparison of HAZUS and model fragility curves for building level connection damage states, for building resistance, $c, 4$	119
Figure E.2	Comparison of HAZUS and model fragility curves for roof cover damage states, for building resistance, $c, 4$	120
Figure E.3	Comparison of HAZUS and model fragility curves for roof sheathing damage states, for building resistance, $c, 4$	120
Figure E.4	Comparison of HAZUS and model fragility curves for opening damage states, for building resistance, $c, 4$	121
Figure E.5	Comparison of HAZUS and model fragility curves for wall damage states, for building resistance, $c, 4$	121

Figure E.6	Comparison of HAZUS and model fragility curves for roof-to-wall connection damage states, for building resistance, $c$ , 4	122
Figure E.7	Comparison of HAZUS and model fragility curves for building level damage states, for building resistance, $c$ , 17	122
Figure E.8	Comparison of HAZUS and model fragility curves for roof cover damage states, for building resistance, $c$ , 17	123
Figure E.9	Comparison of HAZUS and model fragility curves for roof sheathing connection damage states, for building resistance, $c$ , 17	123
Figure E.10	Comparison of HAZUS and model fragility curves for openings damage states, for building resistance, $c$ , 17	124
Figure E.11	Comparison of HAZUS and model fragility curves for wall damage states, for building resistance, $c$ , 17	124
Figure E.12	Comparison of HAZUS and model fragility curves for roof-to-wall connection damage states, for building resistance, $c$ , 17	125
Figure F.1	Flood fragility curves for one-story building with no flood mitigation	127
Figure F.2	Flood fragility curves for one-story building with elevated electrical and appliances	128
Figure F.3	Flood fragility curves for one-story building with upgraded siding and insulation	128
Figure F.4	Flood fragility curves for one-story raised building	129
Figure F.5	Flood fragility curves for two-story building with no flood mitigation	129
Figure F.6	Flood fragility curves for one-story building with elevated electrical and appliances	130
Figure F.7	Flood fragility curves for two-story building with upgraded siding and insulation	130
Figure F.8	Flood fragility curves for two-story raised building	131

## LIST OF TABLES

Table 1	Building category definitions for case study	31
Table 2	Components, component configurations, and component resistances for case study	33
Table 3	Component damage states considered in case study	39
Table A.1	Case study building type design	75
Table B.1	Component configurations	79
Table C.1	Damage state grouping by reconstruction cost	91
Table D.1	Roof zone gust factor and external pressure coefficients	107
Table D.2	Wall gust factor and external pressure coefficients	110
Table D.3	Wall main wind force resistance system external pressure coefficients	113
Table G.1	Cost estimate for model building	133
Table G.2	Estimated Acquisition Costs	134
Table G.3	Roof Cover Cost	135
Table G.4	Roof Sheathing Cost	135
Table G.5	Roof-to-Wall Connections Cost	135
Table G.6	Openings Cost	136
Table G.7	Wall Cost	136
Table G.8	Flood Cost	136
Table G.9	Estimated Mitigation Costs	137
Table G.10	Repairs associated with roof sheathing component damage states	146
Table G.11	Repairs associated with roof cover component damage states	146

Table G.12 Repairs associated with roof-to-wall connection component damage states	147
Table G.13 Repairs associated with opening component damage states	147
Table G.14 Repairs associated with wall component damage states	147
Table G.15 Repairs to building associated with flood component damage states	147

## CHAPTER 1

### INTRODUCTION

Building codes are an effective, but relatively slow mechanism for reducing regional natural disaster risk, since codes generally apply only to new construction and therefore their effect is limited by natural rates of building demolition and replacement. Regional natural disaster risk cannot be fully managed, therefore, without addressing the large portion of the existing building stock that is vulnerable. Recognizing this, a scattered, but increasing number of policies and programs have been implemented to encourage mitigation of existing buildings at the regional scale. The North Carolina Department of Insurance, for example, announced in June 2010 that qualifying homes on the North Carolina coast will receive insurance premium discounts for wind and hail coverage if they undertake certain mitigation efforts. In California, mandatory earthquake retrofit policies have been applied to hospitals, schools and some other public buildings and unreinforced masonry buildings (OSHPD 2005). The Federal Emergency Management Agency (FEMA) has funded grants for mitigation through the Hazard Mitigation Grant Program, Project Impact, the Flood Mitigation Assistance program, the Pre-Disaster Mitigation program, Repetitive Flood Claims program, and Severe Repetitive Loss program. Those grants have been used, for example, to elevate flood-prone homes and businesses, to acquire and demolish or relocate flood-prone homes, and to retrofit buildings to minimize damage from high winds, flooding, earthquakes, and other hazards.



In designing and prioritizing policies and programs like these in an environment of limited budgets and competing objectives, it is difficult to know how to spend funds most effectively and efficiently. This paper introduces a linear program to help guide optimal expenditure of government spending for regional hurricane risk management and to provide insight into some of the complexities involved in designing and prioritizing regional mitigation policies and programs. Specifically, it aims to help answer the questions: 1) How much should be spent on mitigation?; (2) What will the return on that investment be?; and (3) How should mitigation funds be spent (i.e., which buildings should be mitigated, how, and when)? The model considers damage from both high winds and storm surge flooding, and includes a realistic, detailed assessment of the actual risk using a carefully selected set of hurricane scenarios that represents the regional hazard and a component-based damage model. It considers physically realistic mitigation strategies rather than crudely shifting fragility curves to reflect an upgrade, and is computationally tractable for real, regional applications. A case study analysis for residential woodframe buildings in the Eastern half of North Carolina is presented to demonstrate how the model can be implemented for a real region and the type of results it can provide.

Chapter 2 outlines the key challenges in allocating resources for natural disaster risk management. The literature on engineering models in support of resource allocation for natural disaster risk management is summarized in Chapter 3, including discussion of how the model presented here extends that work. Chapter 4 presents the optimization model formulation and solution procedure, and Chapter 5 describes how

the required inputs can be developed. In Chapter 6 the case study results are discussed and Chapter 7 outlines conclusions and future work. Appendices A through G detail calculations and input values used in the models, and Appendices H through K provide the matlab code for the models with accompanying user manuals.

## CHAPTER 2

### THE CHALLENGE OF REGIONAL HURRICANE RISK MANAGEMENT

Determining the most effective and cost-efficient way to reduce risk at a regional scale is complex for many reasons. There are an overwhelming number of possible mitigation alternatives, each with a different cost and effect on the risk. Considering just hurricane risk to residential buildings, for example, one has to decide which of thousands of structures to mitigate, how, and when. For example, one could focus on different subsets of buildings defined by size, location, or structural features (e.g., gable roof). Possible mitigation efforts for each building include elevating or acquiring and demolishing structures to reduce flood damage, and installing hurricane shutters, or roof-to-wall hurricane straps, or gable-end bracing to reduce wind damage. Policies could focus on implementing mitigation efforts before the next hurricane, and/or being prepared to incorporate mitigation as part of the repair and reconstruction that will follow the next event. The problem dimensions expand further because of the many types of impact a hurricane can have, including for example, deaths, injuries, structural damage, business interruption, environmental damage, and induced damage. Moreover, impacts across a region are spatially correlated, which should be considered because it affects the variability of total regional losses. The regional hurricane risk mitigation decision problem is also highly uncertain and dynamic. There is substantial variability in possible mitigation investment outcomes because of the large uncertainty in hurricane occurrence. Because the return periods of damaging hurricanes are

generally decades, a long time horizon is required for analysis. Finally, an appropriate mitigation plan depends on the character of risk in the specific region of interest—what is driving it (e.g., frequent hurricanes, vulnerable structures) and what is “controllable.” Each mitigation strategy targets a different aspect of risk, so the best combination of efforts will be tailored to the issues particular to the region. The model described herein aims to provide insight into some of the complexities involved in designing and prioritizing regional mitigation policies and programs.

## CHAPTER 3

### RESOURCE ALLOCATION FOR NATURAL DISASTER RISK MANAGEMENT

#### ***3.1 Previously Developed Models***

Dodo et al. (2005) summarizes previous research related to resource allocation for natural disaster risk management and groups it into four approaches: Deterministic net present value (NPV) analysis, stochastic NPV analysis, multiattribute utility models, and optimization models. Deterministic NPV (or benefit-cost), the simplest and most common method, involves (1) estimating the cost of implementing each mitigation alternative, (2) estimating the benefits of each mitigation alternative, where benefits are losses avoided by implementing the alternative, and (3) comparing alternatives according to NPV or a similar criterion. Examples include Kappos and Dimitrakopoulos (2008), Altay et al. (2002), and Kuwata and Takada (2003). Stochastic NPV is similar, except instead of a single point estimate, a probability density function of the benefits of each mitigation alternative is estimated, where the uncertainty is due to uncertainty in earthquake or hurricane occurrence (e.g., Englehardt and Peng 1996, Bernknopf et al. 2001, Mostafa and Grigoriu 2002). A few studies have used multiattribute utility theory, such as, Nuti and Vanzi (1998) and Opricovic and Tzeng (2002). Finally, while the first three approaches compare a small number of predefined mitigation alternatives, some have used optimization, in which a set of mitigation alternatives is selected to maximize some stated objective(s) subject to constraints. Shah et al. (1992) performed integer programming to maximize NPV,

considering structural earthquake mitigation alternatives for buildings at Stanford University. Augusti et al. (1994) used dynamic programming to select structural mitigation alternatives for bridges in a highway network in Italy. Researchers at the International Institute for Applied Systems Analysis (IIASA) developed a spatial-dynamic stochastic optimization model to select the insurance policy design (e.g., insurance premiums) that best satisfies decision criteria, such as maximizing insurance company profits (Ermoliev et al. 2000). Versions of the model have been applied, for example, to the cases of earthquake risk in Russia (Amendola et al. 2000) and flood risk in Hungary (Ermolieva et al. 2003).

A few recent studies by two of the advisors of this dissertation have used this optimization approach, which serves as the foundation for the new model described herein. Dodo et al. (2005) developed a linear program to support regional earthquake mitigation resource allocation and illustrated its use with a small case study in Los Angeles County. The model determined which buildings—by structural type, occupancy type, and census tract location—should be upgraded so as to minimize total mitigation and expected post-earthquake reconstruction expenditures. Dodo et al. (2007) presented two efficient solution algorithms to solve the model for a realistic application area. Davidson et al. (2005) extended this model to include the objective of ensuring equity among various groups of people in earthquake risk management. Xu et al. (2007) extended the original model into a stochastic optimization that explicitly models the *variability* in annual earthquake loss (and therefore net benefit of mitigation investments), allowing examination of the risk-return tradeoff between the

original objective of minimizing the sum of mitigation and expected reconstruction expenditures, and a second objective to minimize the chance of an unacceptably large loss in any one earthquake. Vaziri et al. (2010) modified Dodo et al. (2005) so that it can be applied to a developing country in which economic resources are more constrained, damage more widespread, and death tolls much higher than in the U.S. All of these models focus on earthquakes, adopt a regional public sector perspective, and share similar definitions of mitigation alternatives. They share key strengths of considering all possible earthquakes while capturing spatial correlation by using individual earthquake scenarios (as opposed to annual expected ground shaking); considering a very large set of mitigation alternatives; and allowing mitigation investments to occur over time as they do in real life (Dodo et al. 2005). The model presented in this dissertation has these same features in relation to hurricanes rather than earthquakes.

### ***3.2 Comparison with the New Model***

This dissertation adopts the same general approach used in Dodo et al. (2005) and Vaziri et al. (2010), but modifies it for application to hurricane risk using a governmental perspective. This adaptation required five main advances. First, to adequately address hurricane risk, it was necessary to address both damage caused by high winds and that caused by storm surge flooding, therefore the formulation had to be extended to handle these two hazards. Second, unlike earthquake loss models, hurricane wind loss models typically require a more detailed, component-based assessment of damage, so that is what was employed in this study. Each building is

considered to be an assembly of components (e.g., roof cover, roof sheathing, windows) that are most important for resisting high wind loads. The response and damage of each component is estimated separately and together determine the overall building response and damage. Third, whereas in the past, the effect of mitigation has been represented rather crudely as a rightward shift in fragility curves (e.g., Vaziri et al. 2010), in this study physically realistic mitigation strategies are defined and their cost and effect on the expected loss is modeled in detail. Among the mitigation strategies considered, for example, are completely elevating a structure by 24 in. and adding hurricane straps to roof trusses spaced at 24 in. on center. With a component-based damage model, implementation of these strategies can be represented directly and the resulting specific effect on expected loss can be computed. Fourth, as a consequence of the first three changes, the number of variables and coefficients in the already large optimization model increased substantially. Several advances were necessary to address the resulting computational challenge. Approaches had to be developed to identify sets of components, component configurations, mitigation alternatives, and damage states that are comprehensive in their coverage yet as small in number as possible. A new heuristic solution algorithm was developed as well, since although it is a linear program, the size of the model makes it difficult to solve with available solvers. Finally, while the previous model versions adopted a generic public sector approach, in this paper, the government perspective is represented more explicitly, considering how the government might spend funds to support mitigation. General repair and reconstruction expenditures, for example, are no longer assumed to come out of the budget since they are typically paid by the building owner or insurer.



## CHAPTER 4

### OPTIMIZATION MODEL

#### ***4.1 Definitions and Scope***

A building *category*  $m$  is a general category of buildings defined based on obvious architectural characteristics (e.g., number of stories, roof shape, and presence of garage). Each building is considered to be an assembly of different *components*  $n$  that are important in determining hurricane damage (e.g., roof covering, roof sheathing, windows). For each component, a few possible component damage states are defined, and the overall building damage state  $d$  is then defined as a combination of the component damage states. Each component also is assumed to have a few possible *configurations*, and each configuration has an associated resistance. For example, roof cover may be regular asphalt shingles (configuration 1) or high wind shingles (configuration 2), with mean wind resistances of 50 psf and 75 psf, respectively. A building's overall resistance to wind and flood loads is described by its *building resistance*  $c$ , the combination of the resistances of each of the building's components. For example, one building resistance level would be that roof cover is asphalt shingles (configuration 1), roof sheathing is panels fastened with 8d nails at 6 in. on center on the edges and 12 in. on center in the interior (configuration 1), roof-to-wall connections are toe-nail connections at 24 in. on center (configuration 1), openings have hurricane shutters on windows and doors (configuration 4), and walls have gable end bracing (configuration 2). Buildings of the same category and building resistance

are assumed to experience similar damage and loss under hurricane wind and flood loading.

The mitigation alternatives considered are acquiring (and demolishing) or upgrading groups of buildings based on their building category  $m$ , building resistance  $c$ , and geographic location  $i$ . One mitigation alternative is defined as upgrading some number of buildings of category  $m$  in area unit  $i$  from one building resistance  $c$  to another building resistance  $c'$ . Any area unit of analysis can be used without modifying the model formulation. The choice depends on data availability, computational demands, and a desire to choose a unit small enough that the hazard is relatively homogenous within it. Census tract is the area unit of analysis used in the case study, and for clarity, it is used to describe the model formulation.

Note that the optimization model uses building-level variables—resistance levels and damage states that refer to a building as a whole, but those resistances and damage states, and therefore, the mitigation strategies, damage levels, and benefits (losses avoided) are based on component-level analysis, making them much more realistic and allowing them to be mapped directly to physical building characteristics and mitigation strategies. The model is expressed in terms of number of buildings (not floor area), and the variables in the model are considered to be continuous because the hazard is represented as an expected value over the time horizon. In reality, a hurricane event might occur in several isolated years over a long time period, say thirty years. Between those events, pre-event mitigation might occur on an annual

basis; however post-event mitigation and acquisition could not occur as these alternatives require damage from an event. Because the model represents the hazard and the resulting damage as an expected value, the resulting post-event mitigations and acquisitions are similarly modeled. In the model, a fraction of a building may be damaged during a given time period, and that fraction would then be mitigated or acquired. Over a long enough time horizon, the entire building would experience damage and be mitigated or acquired.

Damage due to both high winds and storm surge flooding are considered. Everywhere wind speeds are considered, open terrain 3-second peak gust wind speeds at 10 m are used, but for clarity of presentation, we refer simply to *wind speed*. Flooding due to storm surge inundation only is considered (not inland rainfall). The hurricane hazard is represented as a relatively small set of hurricane scenarios  $l$ , each of which has an associated adjusted annual occurrence probability  $P^l$ , such that together they represent the regional hazard (Section 5.1). In a sense, each scenario represents all hurricanes that would produce similar wind speeds and surge depths in the study area. The annual occurrence probabilities do represent the threat of multiple events during one year. However, the restriction is imposed that between events, all damage is either fixed or the damaged buildings are removed entire from the inventory. That restriction ensures that if a second event strikes, there are no buildings in a weakened state. For each scenario, the wind speed and surge depths are estimated throughout the study area. This model deals with direct loss related to structural and non-structural interior damage, not for example contents loss, indirect business loss or deaths. It also focuses

on structural upgrading and replacement as mitigation alternatives, not insurance, or other types of alternatives. Benefits and costs unrelated to hurricane risk are not considered.

#### 4.2 Modeling Approach

Conceptually, the linear program is formulated to represent the evolving condition of the study area's building inventory, described in terms of the amount of floor area of each building category  $m$  and building resistance  $c$  in each census tract  $i$  (Figure 1). In each time period, decisions are first made about which buildings to mitigate and how. Those decisions are implemented, and then the expected annual damage occurs, calculated as the product of the expected damage given a hurricane and its hazard-consistent occurrence probability, summed over all possible hurricanes. Possible alternatives after a building enters a damage state are to repair it to its pre-hurricane building resistance  $c$ , repair it and simultaneously upgrade it to a better building resistance  $c'$ , or acquire and demolish it. Finally, the state of the building inventory is updated for use in the beginning of the next time period.

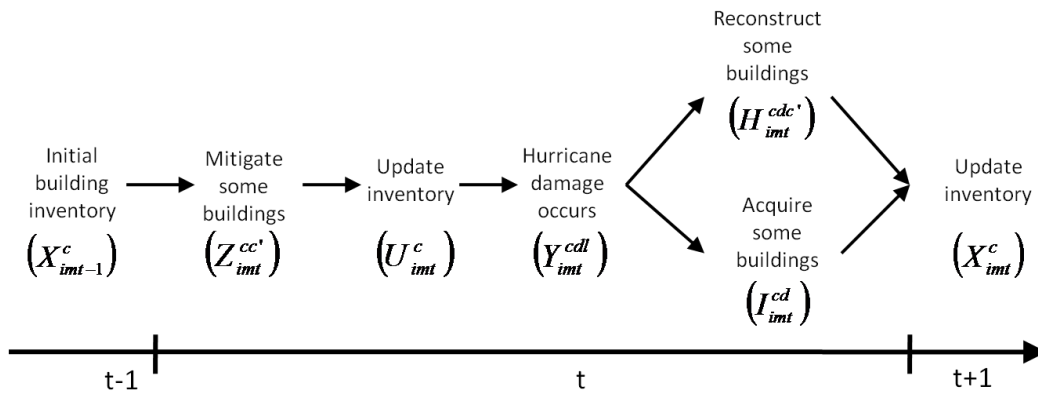


Figure 1. Evolution of building inventory in time period  $t$

### 4.3 Model Formulation

**Mitigation.** Let  $X_{imt-1}^c$  be the number of buildings of building category  $m$  and building resistance  $c$  in census tract  $i$  at the end of time period  $t-1$ . The first decision the model makes is which buildings to mitigate pre-disaster and how. Let  $Z_{imt}^{cc'}$  be the number of buildings of building category  $m$  in census tract  $i$  that are mitigated from building resistance  $c$  to building resistance  $c'$  during time period  $t$ . Mitigation decisions are then represented as:

$$U_{imt}^c = X_{im,t-1}^c - \sum_{c' \in \Delta_c} Z_{imt}^{cc'} + \sum_{c \in \bar{\Delta}_{c'}} Z_{imt}^{c'c} \quad \forall i, m, t, c \quad (1)$$

where  $U_{imt}^c$  is the number of buildings of building category  $m$  and building resistance  $c$  in census tract  $i$  at time  $t$ . So  $X_{im,t-1}^c$  is the inventory at the beginning of the time step, and  $U_{imt}^c$  is the inventory after mitigation decisions have been implemented (Figure 1). The set  $\Delta_c$  identifies for each building resistance,  $c$ , a set of building resistances,  $c'$ , to which the building could be mitigated.  $\bar{\Delta}_{c'}$  identifies for each building resistance,  $c$ , the building resistances,  $c'$ , which could be mitigated to resistance  $c$ .

**Hurricane damage.** Let  $Y_{imt}^{cdl}$  be the number of buildings of building category  $m$  and building resistance  $c$  in census tract  $i$  that are damaged to damage state  $d$  as a result of hurricane  $l$  in time  $t$ . It is calculated as the product of the building floor area (after pre-

event mitigation) and the proportion of buildings of category  $m$  and building resistance  $c$  in census tract  $i$  that will be damaged to damage state  $d$  by hurricane  $l$ ,  $A_{im}^{cdl}$ :

$$Y_{imt}^{cdl} = A_{im}^{cdl} U_{imt}^c \quad \forall i, m, t, c, d, l \quad (2)$$

**Reconstruction and acquisition.** The building inventory damaged in time  $t$  is either reconstructed (and possibly mitigated to a higher component resistance level  $c'$ ), or acquired and removed from the building inventory. Let  $H_{imt}^{cdc'}$  be the number of buildings of building category  $m$  and building resistance  $c$  in census tract  $i$  that are damaged to damage state  $d$  and reconstructed to building resistance  $c'$  in time  $t$ , and let  $I_{imt}^{cd}$  be the number of buildings of building category  $m$  and building resistance  $c$  in census tract  $i$  that are damaged to damage state  $d$  and are instead acquired in time  $t$ . The total inventory that is damaged is equal to the sum of the buildings that are reconstructed and those that are acquired:

$$0 = \sum_l P^l Y_{imt}^{cdl} - \sum_{c' \in \Delta_c} H_{imt}^{cdc'} - I_{imt}^{cd} \quad \forall i, m, t, c, d \quad (3)$$

**Updated inventory.** The building floor area of building category  $m$  and building resistance  $c$  in census tract  $i$  at the end of time period  $t$  is the building floor area at the beginning of the time step, adjusted for pre-event mitigation, subtracting the buildings that are damaged, and adding the buildings that are reconstructed to  $m$  and  $c$ :

$$X_{imt}^c = U_{imt}^c - \sum_l \sum_d P^l Y_{imt}^{cdl} + \sum_{c' \in \bar{\Delta}_c} \sum_d H_{imt}^{c'dc} \quad \forall i, m, t, c \quad (4)$$

**Budget.** An annual budget constraint can be imposed to restrict the amount of money spent on mitigation and acquisition in any given year. In this formulation, we assume there are separate budgets for pre-event spending and post-event spending. Specifically, the total expenditures for mitigating buildings before hurricane damage in one time period  $t$  must be less than a user-defined pre-event budget,  $B_{pre,t}$  (Expression 5), and the total expenditures for mitigating or acquiring buildings as part of reconstruction after hurricane damage in one time period  $t$  must be less than a user-defined post-event budget,  $B_{post,t}$  (Expression 6):

$$\sum_i \sum_m \sum_c \left( \sum_{c' \in \Delta_c} F_m^{cc'} Z_{imt}^{cc'} \right) \leq B_{pre,t} \quad \forall t \quad (5)$$

$$\sum_i \sum_m \sum_c \left( \sum_d \sum_{c' \in \Delta_c} M_m^{cdc'} H_{imt}^{cdc'} + \sum_d V_m^{cd} I_{imt}^{cd} \right) \leq B_{post,t} \quad \forall t \quad (6)$$

where  $F_m^{cc'}$  is the cost per building to mitigate buildings of category  $m$  from building resistance  $c$  to building resistance  $c'$  before an event, and  $V_m^{cd}$  is the cost per building to acquire a building in category  $m$  that are designed to building resistance  $c$  that are damaged to damage state  $d$ . If a building of category  $m$  and building resistance  $c$  is damaged to damage state  $d$ , the cost per building to mitigate it to building resistance  $c'$  as part of the reconstruction is  $M_m^{cdc'}$ . The cost  $M_m^{cdc'}$  is only the cost beyond the cost to reconstruct the building to its original building resistance  $c$ . In this formulation, the

budget constraint reflects the assumption that the government would fund only acquisition and mitigation (either pre-event or post-event in conjunction with reconstruction), not reconstruction of buildings to their original state  $c$ . While repair of home damage is typically paid for by individual building owners or insurers, the government might pay to acquire and demolish particularly high risk homes, or offer grants or incentives for mitigation. Here we assume the government would pay for the entire cost of mitigation; one could easily modify the formulation to reflect an alternate assumption, such as, the government paying some percentage of the mitigation cost.

**Non-negativity constraints.** The following non-negativity requirements must also hold for the decision variables:

$$U_{imt}^c, X_{imt}^c \geq 0 \quad \forall i, m, t, c \quad (7)$$

$$I_{imt}^{cd} \geq 0 \quad \forall i, m, t, c, d \quad (8)$$

$$Z_{imt}^{cc'} \geq 0 \quad \forall i, m, t, c, c' \quad (9)$$

$$H_{imt}^{cdc'} \geq 0 \quad \forall i, m, t, c, d, c' \quad (10)$$

$$Y_{imt}^{cdl} \geq 0 \quad \forall i, m, t, c, d, l \quad (11)$$



**Objective.** The objective of the model is to minimize the total cost of pre-event mitigation, post-event reconstruction, post-event mitigation, and post-event acquisition, as represented by the four terms in the objective function, respectively:

$$\text{Min. } \sum_i \sum_m \sum_t \left( \sum_c \sum_{c' \in \Delta_c} F_m^{cc'} Z_{imt}^{cc'} + \sum_c \sum_d \sum_{c' \in \Delta_c} (R_m^{cdc} + M_m^{cdc'}) H_{imt}^{cdc'} \right. \\ \left. + \sum_c \sum_d V_m^{cd} I_{imt}^{cd} \right) \quad (12)$$

where  $R_m^{cdc}$  is the cost per building to reconstruct a building of category  $m$  to its original building resistance  $c$  after it has been damaged to damage state  $d$ . Note that while the cost to reconstruct damaged buildings is not included in the budget constraint, it is included as part of the model's objective. This corresponds to the assumption that although the government does not pay to repair damaged homes to their original state, it does desire that losses are minimized overall.

The final model is a linear program with the objective in Expression (12) and the constraints in Expressions (1) to (11). Model results indicate how to allocate the annual budgets among pre-event mitigation alternatives  $Z_{imt}^{cc'}$ , and post-hurricane reconstruction alternatives  $H_{imt}^{cdc'}$  and acquisition alternatives  $I_{imt}^{cd}$ , both in terms of dollar expenditures and numbers of buildings modified. They also specify the expected reconstruction expenditures. Section 4.4 presents the model solution procedure and Chapter 5 describes the inputs required and how to obtain them.

#### ***4.4 Solution Procedure***

Even after extensive efforts to reduce its size (see Chapter 5), the optimization model is extremely large since the decision variables all have multiple indexes ( $m$ ,  $c$ ,  $i$ ,  $t$ ,  $d$ , and/or  $l$ ). As an example, the case study application to Eastern North Carolina included more than 25 trillion variables. Thus, while linear programs are generally relatively easy to solve, its size makes this one computationally prohibitive, so a heuristic was developed to solve it. The heuristic starts from the beginning of the planning horizon and moves forward through time. At each time step, as long as funds remain available, it selects the pre-event mitigation alternatives with the highest net benefit per dollar spent, then estimates damage, and finally selects the post-event mitigation and acquisition alternatives with the highest net benefit per dollar spent. Net benefit is calculated as the net reduction in the contribution to the objective function value over the time horizon (i.e., net reduction in total mitigation, acquisition, and reconstruction expenditures). Section 4.4.1 explains how the benefit-cost metric is defined for each type of mitigation alternative, and Section 4.4.2 summarizes the solution algorithm.

##### ***4.4.1 Benefit-Cost Metric Definitions***

The benefit-cost metric is defined in the following slightly different ways for the three types of alternatives—pre-disaster mitigation, post-disaster mitigation, and post-disaster acquisition. The benefit-cost metric for a mitigation or acquisition alternative is always the benefit of the mitigation (i.e., net reduction in the contribution to the objective value), divided by the cost of the mitigation or acquisition. For **pre-disaster**

**mitigation alternatives**,  $\alpha_{imt}^{cc'}$ , the value of the benefit-cost metric to mitigate buildings of category  $m$  in census tract  $i$  from building resistance  $c$  to  $c'$  in time  $t$  becomes:

$$\alpha_{imt}^{cc'} = \frac{(L_{imt}^{cc} - L_{imt}^{cc'})}{F_m^{cc'}} \quad \forall i, m, t, c, c' \in \Delta_c \quad (13)$$

$$L_{imt}^{cc'} = P_{imt}^{cc'} + \sum_{t'=t+1}^T P_{imt'}^{c'c'} \quad \forall i, m, t, c, c' \quad (14)$$

$$P_{imt}^{cc'} = F_m^{cc'} + \sum_d R_m^{c'dc'} (\sum_l P^l A_{im}^{c'dl}) \quad \forall i, m, c, c' \in \Delta_c \quad (15)$$

where  $L_{imt}^{cc'}$  is the sum of the net contribution to the objective value (total costs incurred) from time  $t$  to the end of the time horizon  $T$  when building category  $m$  in census tract  $i$  is mitigated from building resistance  $c$  to  $c'$  in time  $t$ ; and  $P_{imt}^{cc'}$  is the net contribution to the objective value in time  $t$  for mitigating building category  $m$  in census tract  $i$  from building resistance  $c$  to  $c'$  in time  $t$ .

For **post-disaster mitigation alternatives**,  $\gamma_{imt}^{cdc'}$ , the value of the benefit-cost metric to mitigate buildings of category  $m$  and building resistance  $c$  in census tract  $i$  that were in damaged state  $d$  to building resistance  $c'$  in time  $t$  is calculated as:

$$\gamma_{imt}^{cdc'} = \frac{(L_{imt}^{cdc} - L_{imt}^{cdc'})}{M_m^{cdc'}} \quad \forall i, m, t, c, d, c' \in \Delta_c \quad (16)$$

$$L_{imt}^{cdc'} = M_m^{cdc'} + R_m^{cdc'} + \sum_{t'=t+1}^T P_{imt}^{c'c'} \quad \forall i, m, t, c, c' \in \Delta_c, d \quad (17)$$

$$P_{imt}^{c'c'} = \sum_d R_m^{c'dc'} (\sum_l P^l A_{im}^{c'dl}) \quad \forall i, m, c', d \quad (18)$$

where  $L_{imt}^{cdc'}$  is the sum of the net contribution to the objective value (total costs incurred over time) from time  $t$  to  $T$  when building type  $m$  in census tract  $i$  is mitigated from building resistance  $c$  to  $c'$  after the building has been damaged to damage state  $d$  in time  $t$ . The cost of the mitigation is defined as the additional cost incurred in the reconstruction of the building beyond the cost to reconstruct the building to its original building resistance.

For **post-disaster acquisition alternatives**,  $\lambda_{imt}^c$ , the value of the benefit-cost metric to acquire buildings of category  $m$  and building resistance  $c$  in census tract  $i$  in time  $t$  is calculated as:

$$\lambda_{imt}^c = \frac{(L_{imt}^c - V_m^c)}{V_m^c} \quad \forall i, m, t, c \quad (19)$$

$$L_{imt}^c = \sum_t^T P_{imt}^c \quad \forall i, m, t, c \quad (20)$$

$$P_{imt}^c = \sum_d R_m^{cdc} (\sum_l P^l A_{im}^{cdl}) \quad \forall i, m, c, d \quad (21)$$

where  $L_{imt}^c$  is the sum of the net contribution to the objective value (total costs incurred over time) from time  $t$  to  $T$  of the acquisition of building type  $m$  in census tract  $i$  of building resistance  $c$ ; and  $V_m^c$  is the cost to acquire a building of category  $m$  and component resistance  $c$ .

#### **4.4.2 Heuristic Algorithm**

The heuristic, implemented in Matlab (R2009a, The MathWorks, Natick, MA), requires the following seven steps (Figure 2). The heuristic requires the creation of an additional variable,  $w_{im}^c$ , the intermediate inventory available for post-event mitigation or acquisition. For each building category  $m$ , building resistance  $c$ , and census tract  $i$ ,  $w_{im}^c$  is the intermediate inventory. After pre-event mitigation,  $w_{im}^c$  is the inventory that has not yet been mitigated or acquired in the current post-event time period. This variable is cleared and reinitialized in each time period.

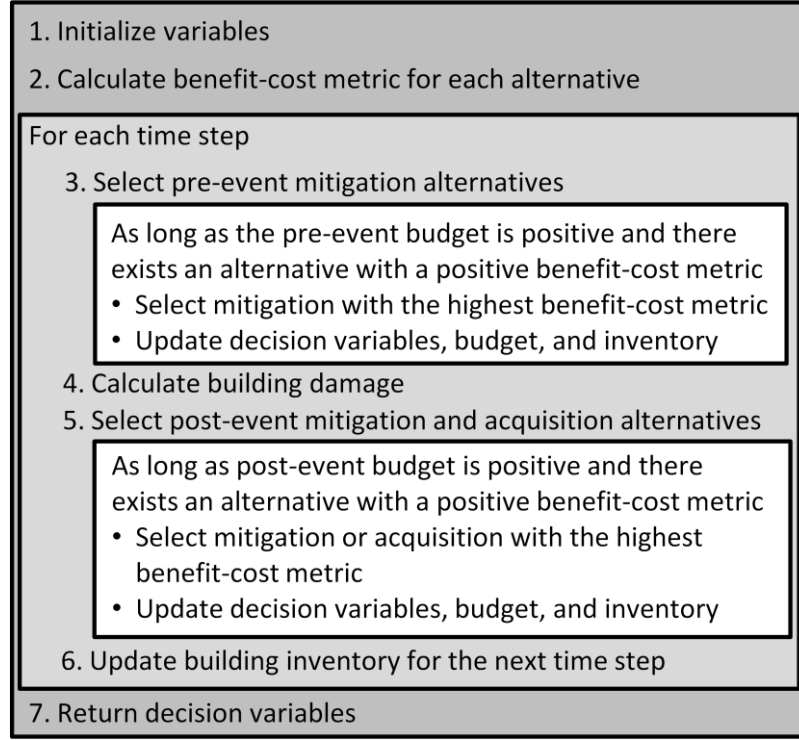


Figure 2. Heuristic solution algorithm

**Step 1. Initialize the solution.** For all  $i, m, t, c, c'$ , initialize all variables such that

$Z_{imt}^{cc'} = 0$ ,  $H_{imt}^{cdc'} = 0$ , and  $I_{imt}^{cd} = 0$ . This corresponds to no mitigation or acquisition for all periods. Notice that this is a feasible solution if there were no damage, but it is infeasible otherwise. Set  $t=1$ .

**Step 2. Calculate benefit-cost metric for each alternative.** Calculate  $\alpha_{imt}^{cc'}$ ,  $\gamma_{imt}^{cdc'}$ , and  $\lambda_{imt}^{cd} \forall i, m, t, c, d, c'$  for time period  $t$

**Step 3. Select pre-disaster mitigation alternatives.** Set  $b = Bpre_t$  and  $w_{im}^c = X_{imt-1}^c \forall i, m, c$ . The pre-event budget and number of buildings for each building

category and building resistance are initialized, and the values are reduced as mitigation funds are spent on pre-event mitigation alternatives.

- a. Select pre-disaster mitigation alternatives to implement. If  $b > 0$ , select the mitigation alternative with the highest non-negative benefit-cost metric,  $PreMit = \max\{\alpha_{imt}^{cc'}\} s. t. \alpha_{imt}^{cc'} \text{ and } w_{im}^c > 0$ . If  $\max\{\alpha_{imt}^{cc'}\} < 0$ , there is no mitigation that can be used to reduce the objective value in this time period, so go to Step 4.
- b. Calculate the number of buildings in the inventory to mitigate. If  $F_m^{cc'} w_{im}^c \leq b$ , the available budget is adequate to mitigate all buildings in census tract  $i$  and building category  $m$  from building resistance  $c$  to  $c'$ . Assign mitigation such that  $Z_{imt}^{cc'} = w_{im}^c$ . If  $F_m^{cc'} w_{im}^c > b$ , the current budget is not sufficient to mitigate all of the buildings. Assign mitigation such that  $Z_{imt}^{cc'} = \lfloor b / F_m^{cc'} \rfloor$ . Only a portion of the buildings in census tract  $i$  and building category  $m$  are mitigated from building resistance  $c$  to  $c'$ .
- c. Update the available budget and building inventory. The budget available in the current period is reduced by the expenditures made in Step 3b, so that the remaining budget becomes:  $b = b - F_m^{cc'} Z_{imt}^{cc'}$ . The building inventory in census tract  $i$ , building category  $m$ , and building resistance  $c$  is updated to reflect mitigations by:  $w_{im}^c = w_{im}^c - Z_{imt}^{cc'} + Z_{imt}^{c'c}$ .
- d. Return to Step 3a until  $b = 0$ , or all mitigation alternatives with a positive return have been examined (e.g,  $\alpha_{imt}^{cc'} < 0 \forall w_{im}^c > 0$ ).

**Step 4. Calculate building damage.**  $w_{im}^c \sum_l (A_{im}^{cdl} P^l) \forall i, m, c, d$ .

**Step 5. Select post-disaster mitigation and acquisition alternatives.** set  $b = B_{post_t}$ . The post-event budget is initialized, and the budget will be reduced as mitigation funds are spent on post-event mitigation alternatives.

- a. Select post-disaster mitigation alternatives to implement. If  $b > 0$ , then select the mitigation or acquisition alternative with the highest non-negative benefit-cost metric,  $Post = \max\{\gamma_{imt}^{cdc'}, \lambda_{imt}^c\}$  and  $w_{im}^c > 0$ . If  $\max\{\gamma_{imt}^{cdc'}, \lambda_{imt}^c\} < 0$ , there is no mitigation that can be used to reduce the objective value, so go to Step 6.
- b. Calculate the number of buildings in the inventory to mitigate or acquire post-disaster.

If the mitigation identified in step 5-a is post-disaster mitigation do the following.

If  $M_m^{cdc'}(w_{im}^c \sum_l (A_{im}^{cdl} P^l)) \leq b$ , the available budget is adequate to mitigate all of the buildings in census tract  $i$  and building category  $m$  from building resistance  $c$  to  $c'$  after they have been damaged to damage state  $d$ . Assign reconstruction-mitigation such that  $H_{imt}^{cdc'} = w_{im}^c \sum_l (A_{im}^{cdl} P^l)$ . If  $M_m^{cdc'}(w_{im}^c \sum_l A_{im}^{cdl} P^l) > b$ , the current budget is not sufficient to mitigate all the buildings. Assign mitigation such that  $H_{imt}^{cdc'} = \lfloor b/M_m^{cdc'} \rfloor$ . Only a portion of the buildings in census tract  $i$  and building category  $m$  are mitigated from building resistance  $c$  to  $c'$  after being damaged to damage state  $d$ .



If the mitigation identified in step a is post-disaster acquisition do the following. If  $V_m^c(w_{im}^c \sum_l (A_{im}^{cdl} P^l)) \leq b$ , the available budget is adequate to acquire all the buildings in census tract  $i$ , building category  $m$ , and building resistance  $c$ . Assign acquisition such that  $I_{imt}^c = w_{im}^c \sum_l (A_{im}^{cdl} P^l)$ . If  $V_m^c(w_{im}^c \sum_l (A_{im}^{cdl} P^l)) > b$ , the current budget is not sufficient to acquire all the buildings. Assign acquisition such that  $I_{imt}^c = \lfloor b/V_m^c \rfloor$ . Only a portion of the buildings of in census tract  $i$ , building category  $m$ , and building resistance  $c$  are acquired and removed from the building inventory.

- c. Update the available post-event budget and building inventory. The budget available in the current period is reduced by the expenditures made in Step 5b, so that the remaining budget becomes:  $b = b - M_m^{cdc'} H_{imt}^{cdc'} - V_m^c I_{imt}^c$ . The building inventory in census tract  $i$ , building category  $m$ , and building resistance  $c$  is reduced by:  $w_{im}^c = w_{im}^c - H_{imt}^{cdc'} - I_{imt}^c$ .
- d. Return to Step 5a until  $b = 0$ , or for all  $w_{im}^c > 0, \gamma_{imt}^{cdc'}, \lambda_{imt}^c < 0$ , then go to Step 6.

**Step 6. Calculate the building inventory for  $t + 1$ :**

$$X_{imt}^c = X_{imt-1}^c - \sum_{c' \in \Delta_c} Z_{imt}^{cc'} + \sum_{c' \in D_c} Z_{imt}^{c'c} - \sum_{c' \in \Delta_c} \sum_d H_{imt}^{cdc'} + \sum_{c' \in D_c} \sum_d H_{imt}^{c'dc} - \sum_d I_{imt}^{cd}$$

Move to the next period  $t = t + 1$  and go to Step 2. If  $t = T$ , then move to Step 7.

**Step 7. Return the decision variables,  $Z_{imt}^{cc'}, H_{imt}^{cdc'}, I_{imt}^c$  for all  $i, m, t, c, d, c'$ .**

#### ***4.4.3 Comparison between heuristic and linear program solver***

To examine the performance of the heuristic solution algorithm, a small test case was solved with both the heuristic and the optimization software OPL Studio (IBM ILOG OPL V6.3) and the results were compared. Because of the size of the optimization, only a very small example could be evaluated in OPL Studio. The test case was for one census tract, one building category, 68 building resistances, 31 damage states, and 97 hurricanes.

Although the nature of the results from the LP and heuristic were different, the results showed a gap of only 0.04%. This does not capture the true gap that would be seen when comparing a larger case study between the two programs. A larger case study would show a larger gap based on the difference in the treatment of pre-disaster mitigation between the LP and heuristic. The LP compares all alternatives for pre-disaster mitigation, post-disaster mitigation, and acquisition together and chooses the best alternatives. When all three alternatives are compared to each other, the post-disaster mitigation and the acquisition alternatives are almost always less expensive than the pre-disaster mitigation option, so pre-disaster mitigation is almost never chosen. The heuristic separates pre-disaster mitigation from post-disaster mitigation and acquisition and requires that within the budget, all pre-disaster mitigation with a positive benefit-cost metric be performed. This results in a higher overall cost. In practice, there might be gaps larger than the gap in this example, however the gap would not be excessive.

## CHAPTER 5

### INPUTS

The mitigation resource allocation optimization model requires several inputs: (a) a set of hurricane scenarios with associated annual occurrence probabilities, (b) a set of building categories, components, and possible component configurations to describe the building inventory, (c) a set of possible building resistances and a set of mitigation alternatives, (d) a description of the initial building inventory in the study area based on those categories and components, (e) a set of component and building damage states, (f) damage estimates for each building and hurricane considering both wind and flood hazards, (g) mitigation, acquisition, and reconstruction costs and budgets. Items (b), (c), and (e) require defining terms, items (a) and (f) require separate analyses, and items (d) and (g) require data collection. As mentioned in Section 4.4, in developing the input, a key concern is the number of variables that result, so the goal throughout was to carefully reduce the number of variables while minimizing the loss of meaning. Each type of input is described in turn in Sections 5.1 to 5.7, including discussion of how the input was developed and the specific input values used for the North Carolina case study. The same processes and in some cases, the same values, could be used to apply the optimization model to another region.

#### ***5.1 Hurricane Scenarios***

Regional hurricane loss models (e.g., HAZUS-MH and FPHLPM) often focus only on the wind hazard, not storm surge flooding, and thus only require modeling regional

wind speeds. They typically use the empirical track model or similar approach to generate tens of thousands of hurricane scenarios and estimate losses for each (e.g., Vickery et al. 2000). In regional mitigation planning it is important to consider the flood hazard as well. In North Carolina, in fact, most state mitigation funds are spent on flood mitigation, not wind mitigation (FEMA 2009, NC DCCPS). Estimation of storm surge inundation is far more computationally demanding than estimation of wind speeds, and thus it cannot be done for each of tens of thousands of hurricane scenarios. It is not computationally tractable to consider thousands of hurricane scenarios in this context in any case since we must evaluate the regional loss associated with each hurricane scenario under each of thousands of mitigation strategies. The resource allocation optimization formulation, therefore, assumes that the hazard can be represented by a relatively small set of hurricane scenarios (on the order of a hundred). Each scenario has an associated hazard-adjusted annual occurrence probability such that together they represent the regional hazard. For each scenario, the wind speeds and surge-related flood depths are computed throughout the study area. The optimization-based probabilistic scenario (OPS) method (Apiratanagul et al. 2010) was used to develop the 97 hurricane scenarios and occurrence probabilities for this case study. For each scenario, the wind speeds and surge-related flood depths can be computed throughout the study area.

This scenario-based approach is necessary to consider both wind and flood damage in an integrated way since both can affect the same buildings in a single hurricane. Separately computing the annual expected wind speeds and annual expected flood

depths at each location, then calculating the annual expected damage due to both together would not capture that interaction. Further, the more general scenario-based formulation allows calculation of a probability density function of annual loss rather than just an expected value, which if desired, would enable straightforward addition of an objective to minimize the chance of an especially large loss (in addition to minimizing expected expenditures), as in Xu et al. (2007) and Vaziri et al. (2009). Although not used in this study, the scenario-based formulation also allows the possibility of employing a damage estimation method based on the time series of wind speeds for each hurricane scenario as in HAZUS-MH (FEMA 2003).

The new optimization-based probabilistic scenario (OPS) method developed by Apivatanagul et al. (2010) was used to develop the hurricane scenarios and occurrence probabilities for this case study. Specifically, the method, which builds on Legg et al. (2010), has 5 main steps (Apivatanagul et al. 2010): (1) developing a large set of candidate hurricane scenarios (defined by full tracks with along-track parameter values); (2) for each scenario, estimating the wind speeds everywhere in the study area and the surge depths at the coastline using a coarse grid for analysis; using a mixed-integer linear optimization to (3) select a reduced set of hurricanes from the candidate set and (4) determine an annual occurrence probability for each scenario so as to match the “true” wind speeds and coastline surge depths as well as possible; and (5) using a fine grid analysis to estimate the inland surge depths everywhere in the study area for each of the scenarios in the reduced set. The result is a relatively small set of hurricane scenarios that together represent the probabilistic regional hazard. For each

of these hurricane scenarios, the method provides an annual occurrence probability and the wind speeds and surge depths throughout the study area. For the case study of Eastern North Carolina, 97 hurricane scenarios were selected (Apivatanagul et al. 2010).

## 5.2 Building Categories, Components, and Possible Component Configurations

Defining the building categories  $m$  that partition the building inventory requires balancing a desire for as few categories as possible to minimize computation with enough categories so that the damage and loss experienced by all buildings within a category will be relatively homogeneous. In the case study, which focuses only on woodframe residential buildings, eight building categories were defined to represent all combinations of number of stories (one or two), garage (yes or no), and roof shape (hip or gable) (Table 1). These were assumed to account for a large portion of the residential buildings in the study region, and to capture the key characteristics important in calculating hurricane loss.

Table 1. Building category definitions for case study

Building category $m$	Stories	Garage	Roof Shape
1	One	Yes	Hip
2	One	Yes	Gable
3	One	No	Hip
4	One	No	Gable
5	Two	Yes	Hip
6	Two	Yes	Gable
7	Two	No	Hip
8	Two	No	Gable

For each category, the basic floor plan and geometry of a model building were defined as necessary to allow detailed component-based damage and loss estimation, and the model building was assumed to represent all buildings in that category. For building category 1, for example, the house has a floor plan of approximately 30 by 60 ft, with 11 windows, 2 doors (one front, one back), a garage door, and a hip shaped roof. Appendix A describes the model buildings for each building category.

Similarly, defining the components to be considered for each building and the possible configurations of each component is a balance between wanting as few as possible, and wanting to include all that are necessary to adequately represent hurricane wind and flood damage. In the case study, six components are considered (Table 2): roof cover, roof sheathing, roof-to-wall connections, openings (i.e., windows, doors, garage doors), walls, and flood susceptibility. The first five are important for determining wind damage; the last one represents the resistance to flood damage. While not a physical building component in the same sense as the others, for purposes of the resource allocation optimization model, flood susceptibility is treated as a component. For each component, the possible configurations (Table 2) were identified so that each is a common physical configuration, and while few in number to minimize computation, together they represent the range of typical resistance levels.

Table 2. Components, component configurations, and component resistances for case study

Component <i>n</i>	Config.	Description	Mean resistance <sup>b</sup>	COV	Reference
Roof cover (RC)	1	Asphalt shingles with 6 nails per shingle	50 psf	0.3	Blue Sky Foundation 2002
	2	High wind shingles	75 psf	0.3	Blue Sky Foundation 2002
Roof sheathing (RS)	1	Panels fastened with corrosion-resistant 8d deformed (ring) shank nails at 6" o.c. along the edge of the panel and at 12" o.c. on the interior supports	60 psf	0.2	Li 2006 Lee 2005
	2	Panels fastened with corrosion-resistant 8d deformed (ring) shank nails at 3" o.c. along the edge of the panel and 6" o.c. on the interior supports	120 psf	0.2	Blue Sky Foundation 2002
	3	Additional panel attachment using a ¼ inch bead of AFG-01 or ASTM D3498 construction adhesive applied to each side of the lumber-to-sheathing joint	225 psf	0.2	APA 2000
Roof-to-wall connections (RTW)	1	Toe-nail connections spaced at 24" o.c.	700 lb	0.2	Conner 1987
	2	Toe-nail connection + structural adhesive spaced at 24" o.c.	1,100 lb	0.2	Reed 1997
	3	Toe-nail connection + Large hurricane strap spaced at 24" o.c.	3,000 lb	0.1	Reed 1997
Openings (O)	1	Standard windows	50 psf	0.3	Blue Sky Foundation 2002
		Standard doors	50 psf	0.3	Blue Sky Foundation 2002
		Standard garage doors	30 psf	0.3	Anderson 2008
	2	Impact resistant/laminated glass (heat strengthened) windows	110 psf	0.3	Norville 2000
		Impact-resistant glass laminated doors	110 psf	0.3	Norville 2000
		Wind reinforced garage doors	90 psf	0.3	FEMA 1993
	3	Plywood shutters on windows	120 psf	0.3	Blue Sky Foundation 2002
		Plywood shutters on doors	120 psf	0.3	Blue Sky Foundation 2002
		Plywood shutters on garage doors	120 psf	0.3	Blue Sky Foundation 2002
	4	Hurricane shutters on windows	175 psf	0.3	Blue Sky Foundation 2002
		Hurricane shutters on doors	175 psf	0.3	Blue Sky Foundation 2002
		Hurricane shutters on garage doors	120 psf	0.3	Blue Sky Foundation 2002
Walls (W)	1	No gable end bracing	2,000 lb	0.2	FDEM 2009
	2	Gable end bracing	3,000 lbs	0.2	FDEM 2009
Flood susceptibility (F)	0	No flood mitigation	n/a	n/a	
	1	Elevated electrical and appliances <sup>b</sup>	n/a	n/a	Taggart 2007, p. 60
	2	Water-resistant foam wall insulation and vinyl siding	n/a	n/a	Taggart 2007, p. 60
	3	Entire house elevated 24 in.	n/a	n/a	Taggart 2007, p. 60

<sup>a</sup> n/a means not applicable<sup>b</sup> Electrical switches and outlets at 60 in (152 cm) above each floor level; electrical panel box at 72 in (183 cm) above lowest floor level; and furnace, air conditioner, water heater, washer, and dryer 18 in (46 cm) above floor level



### ***5.3 Building Resistances and Mitigation Alternatives***

The hurricane wind and flood resistance of each building is defined by the combination of resistances of its components. With two alternatives for roof cover, three for roof covering, four for openings, three for roof-to-wall connections, two for walls, and four for flood, there are theoretically 576 possible building resistances  $c$ . Since a mitigation alternative is defined as a change from one component resistance  $c$  to another  $c'$ , this results in  $(576)^2$ , or almost 332,000 mitigation alternatives. With the goal of defining a set of building resistances such that each is physically realistic and reasonably common, they are substantially different from one another, and together they are relatively few in number while representing the range of possible resistances, the set of 576 building resistances was reduced to 68. It was assumed, for example, that if a homeowner installed expensive high wind shingles, they would install high strength sheathing as well, and therefore the building resistances with high wind shingles and low strength sheathing were omitted. Figure 3 summarizes the building resistances. For example, building resistance level  $c=4$  is defined as having roof cover configuration 1, roof sheathing configuration 2, roof-to-wall connections configuration 3, openings configuration 2, walls configuration 2, and flood susceptibility configuration 1, where the component configurations are defined in Table 2.

Building resistance $c$	Component configurations	Building resistance $c$	Component configurations	Building resistance $c$	Component configurations	Building resistance $c$	Component configurations	Building resistance $c$	Component configurations	Building resistance $c$	Component configurations
1	1,1,1,1,1,1	13	1,3,3,4,2,1	25	1,2,3,3,2,2	37	2,2,1,4,1,2	49	1,3,3,2,2,3	61	2,2,3,3,2,4
2	1,2,2,2,2,1	14	2,2,1,1,1,1	26	1,2,3,4,2,2	38	1,1,1,2,2,2	50	1,3,3,4,2,3	62	2,2,3,4,2,4
3	2,2,2,2,2,1	15	2,2,1,2,1,1	27	2,2,3,3,2,2	39	1,2,2,2,2,3	51	2,2,1,2,1,3	63	1,3,2,2,2,4
4	1,2,3,2,2,1	16	2,2,2,1,2,1	28	2,2,3,4,2,2	40	2,2,2,2,2,3	52	2,2,2,1,2,3	64	1,3,3,3,2,4
5	2,2,3,2,2,1	17	2,2,3,1,2,1	29	1,3,2,2,2,2	41	1,2,3,2,2,3	53	2,2,3,1,2,3	65	1,3,3,3,2,4
6	1,2,3,3,2,1	18	2,2,1,4,1,1	30	1,3,3,2,2,2	42	2,2,3,2,2,3	54	2,2,1,4,1,3	66	1,3,3,4,2,4
7	1,2,3,4,2,1	19	1,1,1,2,2,1	31	1,3,3,3,2,2	43	1,2,3,3,2,3	55	1,2,2,2,2,4	67	2,2,1,2,1,4
8	2,2,3,3,2,1	20	1,1,1,1,1,2	32	1,3,3,4,2,2	44	1,2,3,4,2,3	56	2,2,2,2,2,4	68	2,2,1,4,1,4
9	2,2,3,4,2,1	21	1,2,2,2,2,2	33	2,2,1,1,1,2	45	2,2,3,3,2,3	57	1,2,3,2,2,4		
10	1,3,2,2,2,1	22	2,2,2,2,2,2	34	2,2,1,1,1,2	46	2,2,3,4,2,3	58	2,2,3,2,2,4		
11	1,3,3,2,2,1	23	1,2,3,2,2,2	35	2,2,2,1,2,2	47	1,3,2,2,2,3	59	1,2,3,3,2,4		
12	1,3,3,3,2,1	24	2,2,3,2,2,2	36	2,2,3,1,2,2	48	1,3,3,2,2,3	60	1,2,3,4,2,4		

Figure 3. Summary of the 68 building resistances  $c$  considered in the case study. For each, the component configurations are listed in the following order—roof cover, roof sheathing, roof-to-wall connections, openings, walls, and flood susceptibility; and the component configuration numbers are those defined in Table 2.

Similarly, the resulting set of  $68^2=4624$  possible mitigation alternatives was narrowed down to a set of 124 such that each alternative is reasonably common, they are substantially different from one another, and together they are relatively few in number while representing the range of mitigation strategies that a homeowner might consider (Figure 4). Appendix B details the process by which the final set of 68 building resistances and 124 mitigation alternatives were selected.

#### ***5.4 Initial Building Inventory***

The initial building inventory was taken from HAZUS-MH MR1 general building stock inventory, which is generated using the publically available 2000 U.S. Census total building inventory statistics (ERSI 2006). HAZUS-MH provided number of one- and two-story residential woodframe buildings in each census tract. To determine the number of buildings in each building category  $m$  in each census tract, it was assumed that 15% of buildings have no garage and 85% of buildings have some kind of garage (NAHB 2004); that 36% of buildings have hip roofs and 64% have gable roofs (Davidson and Rivera 2003); and that the attributes number of stories, garage, and roof shape occur independently. There are a total of 755,948 residential wood frame buildings in the initial building inventory.

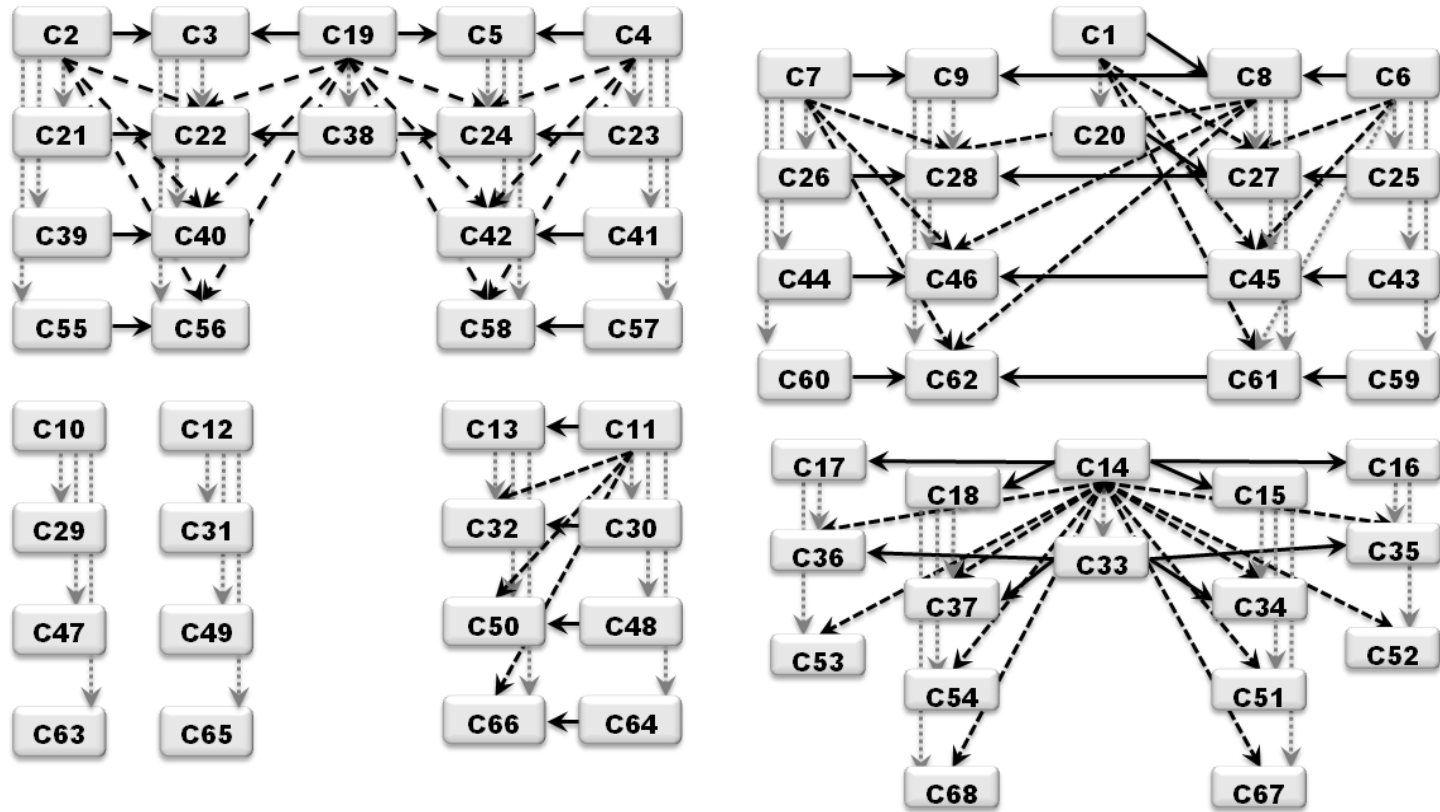


Figure 4. Summary of the 124 mitigation alternatives considered in the case study. Each rectangle C1 to C68 is a building resistance. Each arrow represents a mitigation alternative, changing from one building resistance to another. A dotted gray arrow represents a mitigation alternative to reduce flood damage, a solid black arrow represents a wind mitigation alternative, and a dotted black arrow represents a mitigation alternative for both flood and wind.

The initial building inventories in each building category were then divided further into building resistances. The initial building inventory was assumed to have no flood resistance (i.e., flood susceptibility configuration 1 in Table 2). In the absence of better data, the initial inventory was then divided evenly among the remaining 19 building resistances,  $c=(1, \dots, 19)$  for each building category in each census tract.

### ***5.5 Component and Building Damage States***

The optimization model assumes that each building can experience one of a number of building damage states  $d$  as a result of the high wind speeds and flood depths associated with a hurricane. Similar to the building resistances, building damage states were defined to be few in number, while together capturing the range of damage levels. Like the building resistances, they are component-based to allow more accurate representation of hurricane impacts and the effects of mitigation efforts. In the case study, damage states for the wind-related components ( $n=RC, RS, RTW, W, O$ ) were defined the same way they are in HAZUS-MH (FEMA 2003) (Table 3). The flood component damage states were defined directly in terms of percentage of building value rather than in terms of qualitative damage.

Table 3. Component damage states considered in case study

Damage state	Roof cover (RC)	Roof sheathing (RS)	Roof-to-wall connections (RTW)	Openings (O)	Walls (W)	Flood susceptibility (F)
0	$\leq 2\%$ failed	No damage	No damage	No opening failed	No damage	None
1	$> 2\%$ and $\leq 15\%$ failed	$\leq 2\%$ failed		1 opening		Minor. 1% to 10% building value
2	$> 15\%$ and $\leq 50\%$ failed	$> 2\%$ and $\leq 5\%$ failed		$> 1$ opening and $\leq 20\%$ (or 3) openings		Moderate. 11% to 20% building value
3	$> 50\%$ failed	$> 5\%$ and $\leq 25\%$ failed		$> 20\%$ (or 3) and $\leq 50\%$ openings		Substantial. 21% to 30% building value
4		$> 25\%$ failed	Totaled	$> 50\%$ openings	Totaled	Severe. 31% to 40% building value

With four damage states defined for roof cover, five for roof sheathing, two for roof-to-wall connections, five for openings, two for walls, and six for flood, considering all possible combinations would result in 2400 possible building damage states. As with building resistances, that set was drastically reduced to 31 by removing those that are unlikely to occur in reality, and by collapsing damage states that result in similar levels of loss since that is ultimately what is important for the resource allocation optimization model. Building damage state  $d=1$  corresponds to no damage for any component,  $d=31$  corresponds to loss of 100% of the building value, and  $d=2$  to  $d=30$  are damage states of increasing severity in between. Building damage state  $d=2$ , for example, corresponds to a situation in which roof cover, roof sheathing, roof-to-wall connections, walls, and flood are all in damage state 0, and openings are in damage

state 1 or 2. Appendix C details the definitions of the final set of 31 building damage states and the process by which it was developed.

### 5.6 Damage Estimates

One of the key inputs to the resource allocation optimization model is the proportion of buildings of category  $m$  and building resistance  $c$  in census tract  $i$  that will be damaged to building damage state  $d$  by hurricane  $l$ ,  $A_{im}^{cdl}$ . To calculate these values for every  $m$ ,  $c$ ,  $l$ ,  $i$ , and  $d$ , detailed component-based damage estimation models were developed considering both wind and flood damage. For each building category  $m$  and building resistance  $c$ , the wind damage model is used to estimate the joint probability of the wind-related component damage states  $d_{RC}$ ,  $d_{RS}$ ,  $d_{RTW}$ ,  $d_O$ , and  $d_W$  as a function of wind speed  $w$ . The hurricane scenario input provides the wind speed  $w$  at each census tract  $i$  in each hurricane scenario  $l$ . They can be combined to provide the probability of each wind-related component damage state for each census tract  $i$  and hurricane  $l$ . Similarly, the flood damage model is used to estimate the probability of each component damage state  $d_F$  for the flood component. Again, using the hurricane scenario input provides the surge depth experienced at each census tract  $i$  in each hurricane  $l$ , allowing translation to the probability of each flood component damage state for each census tract  $i$  and hurricane  $l$ . Finally, assuming that the probability of each flood damage state is independent of the joint probability of the wind-related component damage states, all six component damage states are used to determine  $A_{im}^{cdl}$ , the proportion of buildings in each *building* damage state  $d$ , for each  $m$ ,  $c$ ,  $l$ , and  $i$ .

Sections 5.6.1 and 5.6.2 describe the methods used to estimate the component damage states for wind and flood components, respectively.

#### ***5.6.1 Wind Damage Model***

The wind damage model was based on the damage model for the Florida Public Hurricane Loss Model (Pinelli, FPHLM 2005), with the exception of the final damage state definitions in Table 3, which were based on HAZUS-MH damage states (FEMA 2003). The Florida Public Hurricane Loss Model was chosen as a model for this wind damage model because it defines a process for estimating wind damage based on minimal storm information. The wind damage model used by HAZUS uses similar formulas to estimate load and damage (from ASCE-7), but takes into account the progression of a storm over time when estimating damage. The advantage of the FPHLM method was the ability to estimate damage based on the maximum wind gust of a storm in a given census tract.

For each building category  $m$ , building resistance  $c$ , and wind speed  $w$ , the wind damage estimation model simulates the joint probability distribution for the component damage states  $d_{RC}$ ,  $d_{RS}$ ,  $d_{RTW}$ ,  $d_O$ , and  $d_W$  (the five wind-related components—roof covering, roof sheathing, roof-to-wall connection, openings, and walls) (Figure 5).



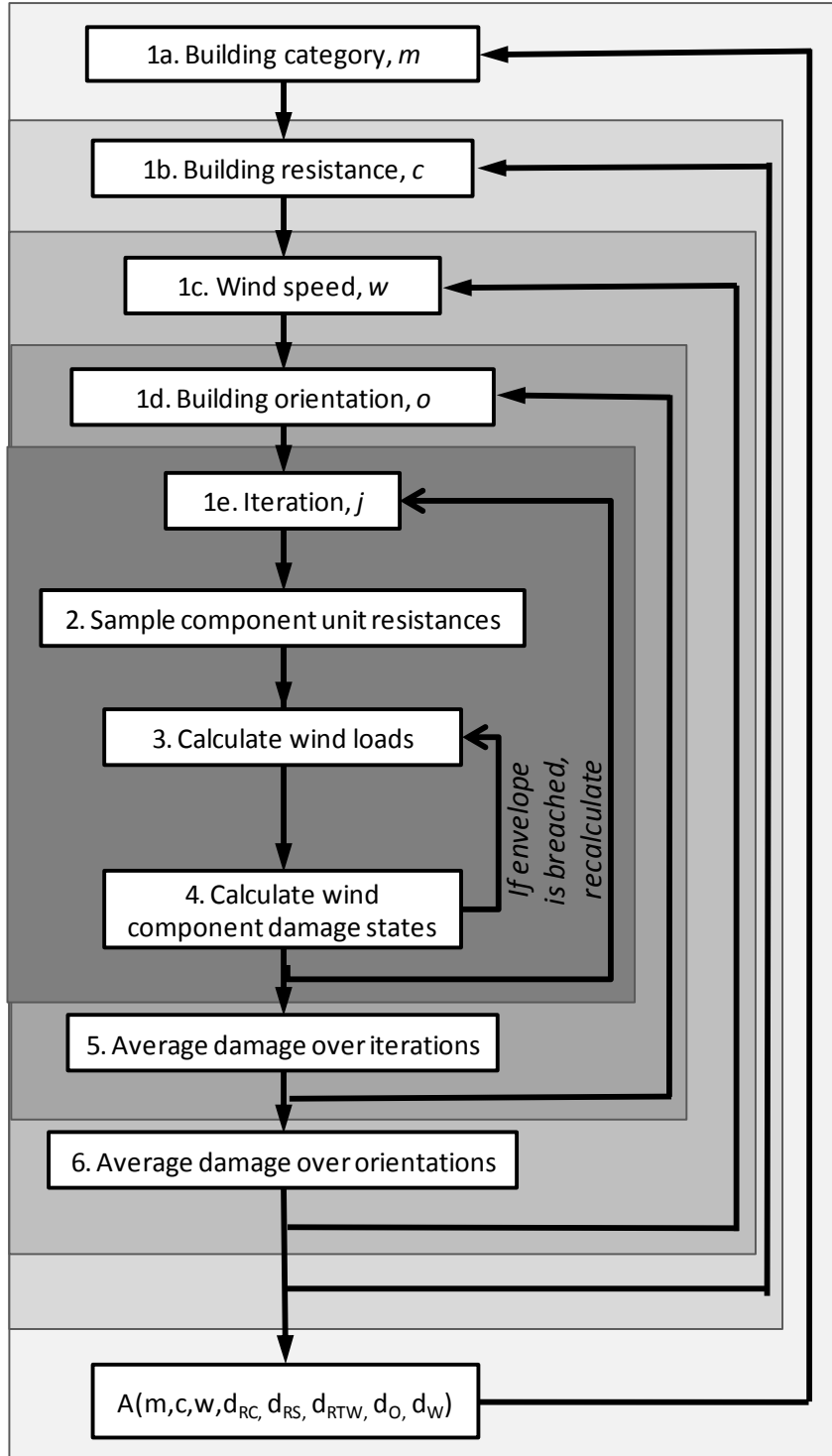


Figure 5. Schematic of wind damage estimation model

First, the building category  $m$ , building resistance  $c$ , wind speed  $w$ , building orientation  $o$  relative to the wind direction, and iteration  $j$  are initialized. Building orientation can significantly affect the wind loads and therefore the damage experienced, so eight different equally-spaced building orientations are considered (Figure 6). Second, while all buildings in the same building category are assumed to have the same layout, in each iteration, the resistances of the specific component units are randomly sampled. The resistance for each component configuration is assumed to be normally distributed with the mean resistance and coefficient of variation given in Table 2, but each unit of each component is sampled individually. For example, on a given hip roof with roof sheathing of configuration type 1, there may be more than 20 roof sheathing panels, and each is sampled individually from the same normal distribution with mean 60 psf and coefficient of variation 0.2. Similarly, the resistance of each window is sampled individually from the same distribution. To ensure there no negative values are assigned, all sampled values must be within two standard deviations of the mean.

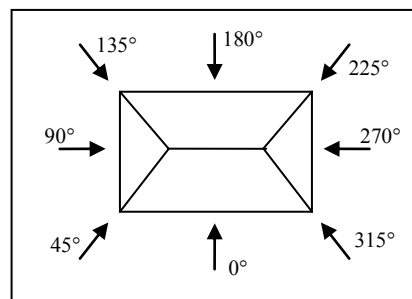


Figure 6. Eight building orientations relative to the wind direction

Third, the wind load on each component unit is calculated as in ASCE-7, similar to the Florida Public Hurricane Loss Model Project (FPHLM 2005). The loads depend on

the wind speed and the orientation of the building relative to the wind. See Appendix D for a detailed description of the load calculations. Fourth, for each component unit, if the resistance is less than the load, it is assumed to have failed. An initial damage check is run to evaluate the extent of damage to the components. If a building's envelope has been compromised, internal pressures develop in the building and the loads applied to the roof, windows, and walls increase, which can result in a dramatic increase in damage to that building. If the initial damage check indicates a damaged building envelope, therefore, the model recalculates the loading and building damage, assuming the increased load resulting from the building envelope breach. The final building damage in terms of, for example, number of roof sheathing panels and windows damaged, is translated into the component damage states defined in Table 3.

Fifth, Steps 2 to 4 are repeated for many iterations (100 iterations were used in the case study), and the component damage is averaged over iterations. Sixth, Steps 2 to 5 are repeated for each of the eight building orientations, and the results are averaged over all building orientations. to produce  $A(m, c, w, d_{RC}, d_{RS}, d_{RTW}, d_O, d_W)$ , the joint probability distribution that a building of category  $m$  with building resistance  $c$  experiences wind-related component damage states  $d_1$  to  $d_5$ , as the result of hurricane wind speed  $w$ .

As discussed above, the wind damage model was based primarily on the Florida Public Hurricane Loss Model's damage estimation model. However, the wind damage model used damage states derived from HAZUS-MH, as well as similar component

strengths and wind loading calculations as used in HAZUS-MH (based on ASCE-7). Therefore, the fragility curves generated by HAZUS-MH were used to validate the results from the wind damage model. Direct comparisons of fragility curves can be found in Appendix E. The fragility curves were compared by estimating damage to similar buildings under similar conditions. The resulting building fragility curves are similar in magnitude and slope for the wind damage model and HAZUS. The wind damage model curves are steeper, i.e. the number of buildings affected by a particular damage state escalates over a shorter change in wind speed. This difference can be primarily attributed to the storm definitions between the two models. The wind damage model takes only one storm variable into account in the damage model, the maximum wind gust. HAZUS estimates damage based on several storm variables that define the character of the wind strength. As a result, HAZUS results have a wider distribution of damage states from a given wind speed, and a more shallow fragility curve.

It should be noted that HAZUS fragility curves are based on lognormal distributions. There is some question as to whether the use of lognormal distributions represents reality accurately. There are some efforts to examine and authenticate the use of the log normal distributions (Gencturk et al 2008, Polese et al 2008). This damage estimation model developed for this model could easily be modified to use other types of distributions.

### ***5.6.2 Flood Damage Estimation***

For each building category  $m$ , building resistance  $c$ , and flood depth  $f$ , the probability that component damage state  $d_F$  occurs was obtained directly from flood fragility curves for woodframe residential buildings. The flood fragility curves (probability of at least damage state  $d_F$  vs surge depth) were developed using results of a flood loss simulation model for residential woodframe buildings described in Taggart (2007). Appendix F details how the fragility curves were developed and presents the fragility curves used.

### ***5.7 Mitigation Costs, Acquisition Costs, Reconstruction Costs, and Budgets***

**Mitigation costs.** The cost associated with each pre-event mitigation option was estimated using R.S. Means (2009) for the wind mitigation options and R.S. Means (2009), FEMA's Homeowners Guide to Retrofitting (2009 b), and the Federal Alliance for Safer Homes (FLASH 2010) for flood mitigation options. For example, the cost to upgrade from standard windows to impact resistant windows would be the cost to replace a window (including material, labor and overhead), multiplied by the number of windows replaced. The cost to mitigate a building post-event was defined as the reconstruction cost beyond the cost to repair the component to its original state. If a component could be mitigated at reconstruction for the same cost as repairing to its original state, then the cost to mitigate was \$0. For example, if the entire roof required replacement, then the difference in the cost to increase the number roof sheathing fasteners would be negligible, and the cost to mitigate would be \$0. However, if one window was broken, but all windows were to be replaced and

upgraded, the mitigation cost would be the cost to replace all windows, minus the cost to replace the window that was broken (which is considered a reconstruction cost).

**Acquisition costs.** An initial building value for each building category,  $m$ , was estimated using R.S. Means (2009). The cost to acquire a building was defined as 110% of the cost to fully reconstruct the building. For example, if a building was demolished and would cost \$100,000 to reconstruct, then acquiring the building would cost \$110,000. The increased acquisition cost was meant to capture some of the related cost to society from the removal of a building from the inventory, including for example, reduction in tax revenue and cost to subsequently maintain the land if it becomes open space. When homes are acquired, the land is often turned into reserve land for the state, which then costs the state to maintain.

**Reconstruction costs.** The cost to reconstruct each component at each damage state was also estimated using R.S. Means (2009). Depending on the level of damage to the component, the element could either be repaired or replaced. For example, it was assumed the entire roof must be replaced if over 15% of the roof sheathing area is damaged. Appendix G summarizes the final cost coefficients that were used in the case study.

**Budgets.** Base case pre- and post-event annual budgets were each assumed to be \$8 million. From 2006 to 2010, North Carolina received \$1 to \$5 million annually from FEMA's Pre-Disaster Mitigation Grant Program (FEMA 2009), and approximately

\$12 million annually from FEMA's Hazard Mitigation Grant Program (North Carolina 2010). Additionally, in the event of a major hurricane, the federal government allocates funds to aid in recovery, including in the reconstruction of residential wood frame buildings. Based on these allocations, an annual budget of approximately \$15 to \$16 million for hurricane mitigation was considered reasonable. The budget was split equally between pre-event and post-event budgets, with \$8 million each. Several other annual budgets were evaluated as well to examine the effect of budget, always assuming pre-event and post-event annual budgets were equal: \$1 million, \$2 million, \$5 million, \$20 million, and infinite.

## CHAPTER 6

### CASE STUDY

#### *6.1 Scope and Overview*

The Eastern half of North Carolina was used as a case study area to develop and demonstrate the new resource allocation model (Figure 7). The area includes the part of the state with the most severe hurricane hazard, extending westward to include half of Raleigh, the state capital. Fifty percent of the state's land area, 506 of the 1555 census tracts, and about 500 km of coastline are included in the study region. It is a low-lying area, with elevations mostly less than 45 m, and no greater than 90 m.

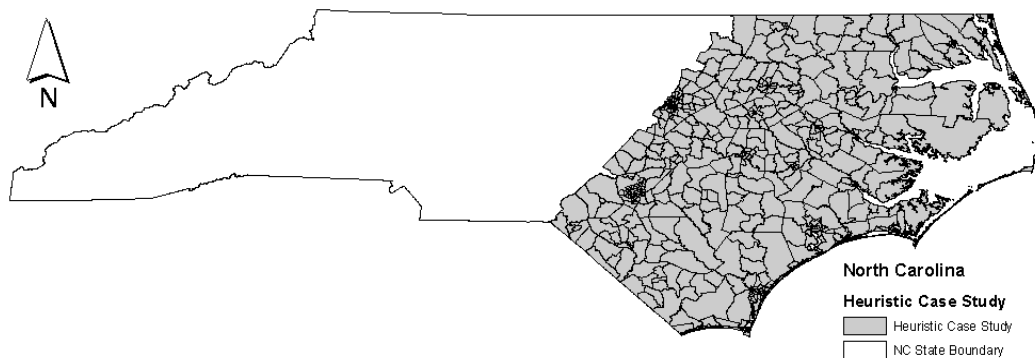


Figure 7. Eastern North Carolina study area with census tract outlines

North Carolina has had a long and destructive history of hurricanes. A tropical storm or hurricane is expected to make landfall on the North Carolina coast on average every four years, and one is expected to pass through the state on average every 1.3 years (SCONC 2010). Recent hurricanes making landfall in North Carolina include Bertha



(1996), Fran (1996), Bonnie (1998), Dennis (1999), Floyd (1999), Isabel (2003), and Charley (2004).

The study focused on residential woodframe buildings only. It uses a one-year time period and a time horizon of 30 years. To avoid an artificial end-of-time-horizon effect, the benefit-cost metric was evaluated for a time horizon of 60 years, but only the first 30 years were considered the investment horizon. So in the first year of the investment horizon, the benefit is calculated for that year and the subsequent 59 years. In the second year, the benefit is calculated for that year and the subsequent 58 years. All other input data was as detailed in Chapter 5.

The model output indicates how to allocate the annual budgets among pre-event mitigation alternatives, and post-hurricane mitigation alternatives and acquisition alternatives, both in terms of dollar expenditures and numbers of buildings modified. Specifically, it recommends expenditures by building category  $m$ , census tract  $i$ , time period  $t$ , and mitigation alternative  $cc'$ . It also includes the expected reconstruction expenditures that result. This huge amount of output can be examined in different ways to answer three main questions of interest to risk managers and policy makers: (1) How much should be spent on mitigation?; (2) What will the return on that investment be?; and (3) How should mitigation funds be spent (i.e., which buildings should be mitigated, how, and when)? Each of those topics is discussed in turn in Sections 6.2 to 6.4.

In examining the case study results, it is important to note the appropriate way to interpret the optimization model. The model is not intended to suggest a strategy of the form, “If Hurricane  $X$  happens at time  $t_1$ , do this; if Hurricane  $Y$  happens at time  $t_2$ , do that.” With an infinite number of possible sequences of events, it would be impossible to develop strategies conditional on every eventuality. Instead, what the optimization does is to recommend how funds should be spent on average, knowing the set of hurricanes possible, the annual occurrence probabilities of each, and the losses that would result from each under various mitigation plans. Specifically, for pre-event decisions, it is unknown which hurricanes will actually occur and when, so decisions are made based on the expected values of costs and benefits. Similarly, after a hurricane occurs, what will happen from that time forward is unknown, so post-event decisions are made based on the expected damage that will occur and the expected values of costs and benefits from that time forward.

## ***6.2 How Much to Spend***

For the case study analysis, the model indicates that it is cost-beneficial to spend on both pre-event mitigation and post-event mitigation and acquisition. For the base case analysis, with annual pre- and post-event annual budgets of \$8 million each, the model recommends spending all of the pre-event funds for the first 16 years and 72% of it in Year 17, and all of the post-event budget every year, resulting in total mitigation and acquisition expenditures of \$374 million over the 30-year time horizon.

Rerunning the model with multiple annual budget levels shows that if more funds are available for mitigation and acquisition, more should be spent. There is, however, a limit to the amount recommended for both pre-event and post-event spending. Figure 8 shows how the (a) recommended pre-event mitigation expenditures, and (b) recommended post-event mitigation and acquisition expenditures over the 30-year time horizon vary with budget. It indicates that even if an infinite pre-event budget was available, only \$224 million should be spent on pre-event mitigation. In fact, the recommended pre-event spending levels off at just \$134 million, at the base case pre-event annual budget of \$8 million. Above that level of spending, few additional cost-effective pre-event mitigation alternatives are available. While additional mitigation could certainly be done (i.e., vulnerable homes remain), the model suggests that it would not be cost-effective to undertake them.

The recommended expenditures for an infinite budget suggest that there is a limit to the post-event expenditures recommended as well, but it is about 25 times higher than the pre-event spending limit. This is because post-event acquisition is far more expensive than pre-event mitigation alternatives, so there is more spending that can be done. The average pre-event mitigation cost for all mitigation alternatives is \$37,500, while the average cost to acquire a home is around \$202,300. In reality, political and other constraints will probably limit the number of acquisitions that can really be undertaken, since each one requires obtaining the consent of the owner, removes the building from the tax base, and requires that the community then maintain the land.

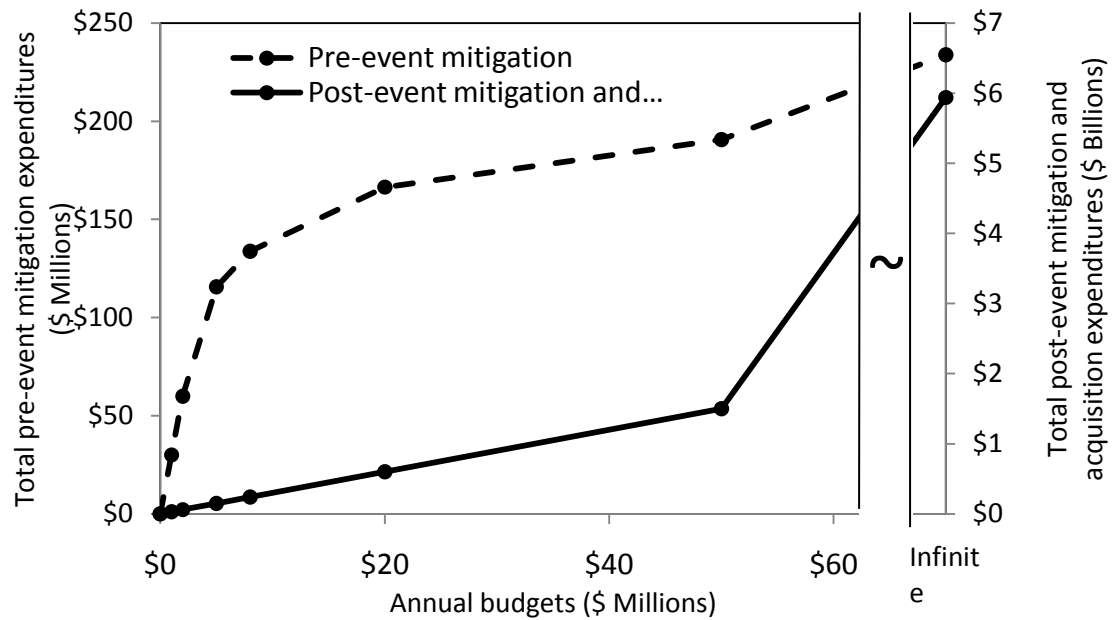


Figure 8. Recommended expenditures vs. annual pre- and post-event budgets, assuming pre- and post-event budgets are equal. *Pre-event* refers to expenditures on pre-event mitigation, *Post-event* refers to expenditures on post-event mitigation and acquisition.

### 6.3 What you get for the Investment

A key question asked in relation to regional hurricane mitigation is what the benefit of an investment would be. An inability to clearly answer that question can create a major impediment for undertaking investments. In fact, in 1999, wanting to better understand the return on the investment of \$3.5 billion of federal and state/local matching funds on flood, windstorm, and earthquake risk mitigation from mid-1993 to mid-2003, the U.S. Congress directed the Federal Emergency Management Agency (FEMA) to fund an independent study to assess the future savings resulting from mitigation activities (Rose et al. 2007). Based on analysis of a database of actual mitigation grants made during that 10-year period, the resulting study concluded that the overall benefit-cost ratios for flood and wind project grants were 5.1 and 4.7,

respectively (Rose et al. 2007). The total cost of those flood and wind project grants over 10 years was \$2.5 billion, or roughly \$250 million per year.

This study provides a way to answer that question *a priori*, comparing the benefit-cost ratios of alternative possible regional investment strategies as part of the decision process during which mitigation investments are designed and selected. Let the benefit be the reduction in expected reconstruction expenditures due to the investment in mitigation and acquisition, and let the benefit-cost ratio be that benefit divided by the total mitigation and acquisition expenditures. For the base case analysis then, with pre- and post-event annual budgets of \$8 million, the benefit is \$2.23 billion and the benefit-cost ratio is 6.0.

Figure 9 shows (a) the overall benefit-cost ratio and (b) the expected total expenditures (mitigation, acquisition, and reconstruction) versus annual budgets. It shows how much the benefit-cost ratio and total expenditures can vary, depending on the investments made. It suggests that at annual pre- and post-event budgets of \$1 million, the benefit-cost ratio is an impressive 19.8 and the expected total expenditures (i.e., the overall cost to society) is \$8,47 billion. The total expenditures assuming \$0 spend on mitigation and acquisition is \$9,59 billion. As the annual budgets increase to \$8 million, the benefit-cost ratio declines steeply to 6.0, and the total expenditures levels off at \$7,74 billion. Note that these results assume spending is undertaken according to the model recommendations, in which the alternatives with the highest return on investment are undertaken first, and those with a lower but still positive return on

investment are done only if funds are available. The important implication is that, if spending is undertaken systematically as suggested by the model, most of the benefit can be obtained with a relatively small budget. This information may help combat a second possible impediment to undertaking regional mitigation efforts—being overwhelmed by the magnitude of the problem and therefore not wanting to do anything. If a community understands that a relatively small amount of funds smartly invested can substantially address the challenge, they may be more willing to start the process. Having quantitative estimates of the costs and benefits of different regional mitigation strategies also can be critical for evaluating proposed mitigation policies in the context of other societal demands, such as, crime or health care.

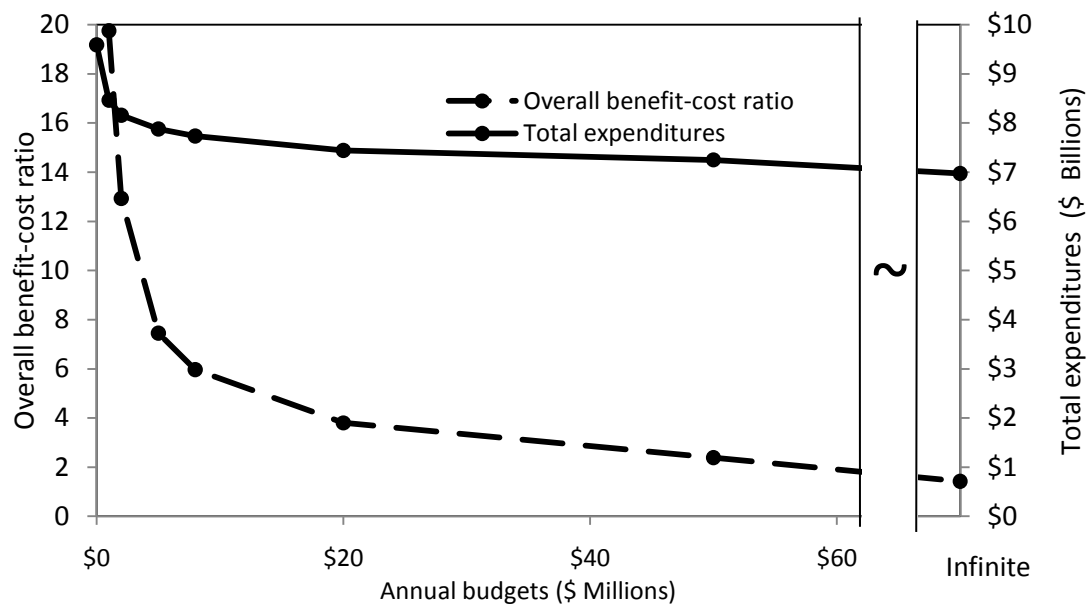


Figure 9. Overall benefit-cost ratio and total (mitigation, acquisition, and reconstruction) expenditures vs. annual pre- and post-event budgets, assuming pre- and post-event budgets are equal.

Finally, Figure 9 shows that even with an infinite budget, the benefit-cost ratio remains above one (1.4) and the total expenditures do not go to zero. This is because the model selects only alternatives that are cost-effective even if additional funds are available. The goal is not zero risk, which would be prohibitively expensive, but an efficient, cost-effective use of funds.

Figure 10 illustrates how the annual reconstruction expenditures decline over time for specific annual budgets. The computations in Figure 9 for total cost only consider the expenditures over a 30 year planning horizon whereas Figure 10 illustrates that there are benefits in these investments which continue into the future. For example, with a \$1 M investment, the annual expenditures for reconstruction drop by about \$50 by the end of the 30 year planning horizon. For an annual budget of \$50 M the decline is over 50% by the end of 30 years.

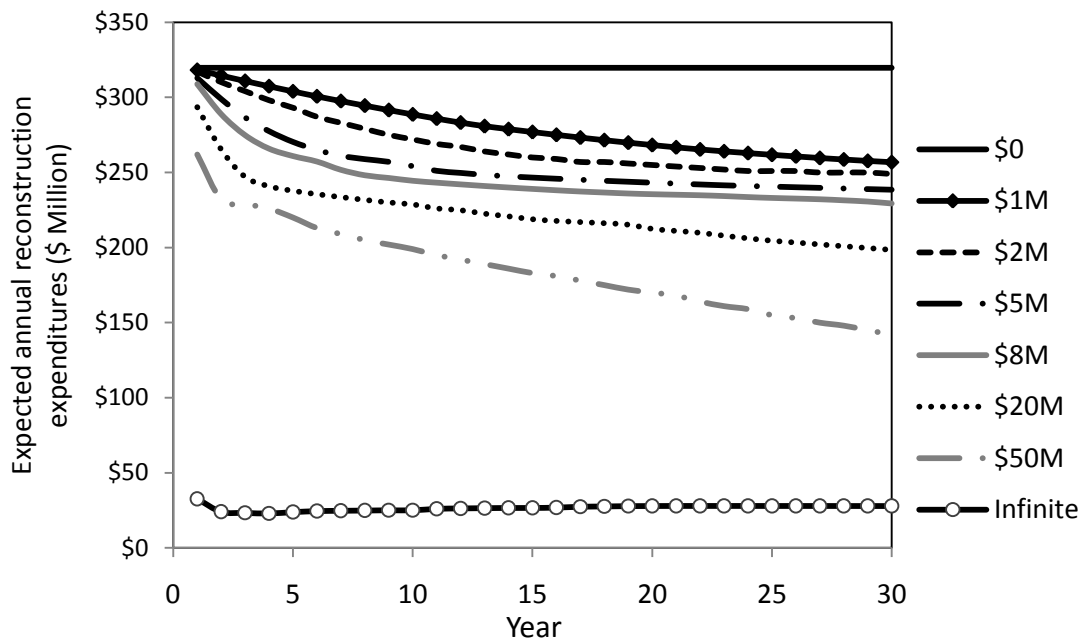


Figure 10. Expected Annual Reconstruction Expenditures for specific mitigation and acquisition annual budgets

#### 6.4 How to Spend It

The third primary use of the model is to help determine *how* to spend funds in a systematic way so as to meet the objective of minimizing total expected expenditures. The best allocation of mitigation and acquisition funds depends on the relative levels of several attributes of the many buildings that may be targeted—the wind and flood hazards to which they are exposed, their current levels of vulnerability, the relative costs of the possible mitigation alternatives, and the effectiveness of those alternatives (i.e., how much they would reduce losses). While some mitigation alternatives can be considered low-hanging fruit (i.e., they are clearly going to be cost-effective), in general it is extremely difficult, if not impossible, to determine how these forces interact to make some mitigation alternatives more appealing investments than others.



The model results can be disaggregated in several ways to gain a better understanding of how to guide regional hurricane risk management spending. Recommended investments are considered by: (a) type of investment (pre-event mitigation, post-event mitigation, or acquisition); (b) hazard they target (wind, flood, or both); (c) when they should be made; (d) geographic location, and (e) specific mitigation strategy (e.g., elevate house, add hurricane shutters). It is important to keep in mind, however, that often the combinations of these features are what determines the appeal of a mitigation alternative.

#### ***6.4.1 By Type of Investment and Targeted Hazard***

The optimization model allows government funds to be spent on 3 main types of investment—pre-event mitigation, post-event mitigation, or acquisition, and at the highest level of aggregation, each mitigation strategy can be considered to target flood damage, wind damage, or both. Figure 11 shows the breakdown of the total government expenditures during the 30-year time horizon, by investment type and targeted hazard, in terms of dollars spent (Figure 11a) and number of buildings affected (Figure 11b). In terms of dollars spent, the model recommends mostly spending on acquisition (51%) and pre- and post-event mitigation of flood damage (40%), with wind-related mitigations making up only 9%. In terms of number of buildings affected, however, pre- and post-event mitigation of flood damage dominate (80%), wind and wind and flood mitigations make up 18%, and acquisition accounts for only 2% of the investment. This difference highlights the differences in the character of the main types of risk reduction strategies. Acquisition is expensive and

focused on very few high risk buildings, while mitigation alternatives directed at only wind or flood damage require fewer funds spread over more buildings. For the \$8 million annual budgets, for example, the model recommends spending \$162,600, \$2600, \$3100, and \$4800 per building, respectively, for acquisition, wind only, flood only, and wind and flood mitigation alternatives. A policy may be easier to implement if focused on fewer buildings, but equity concerns may increase the appeal of policies that help more buildings.

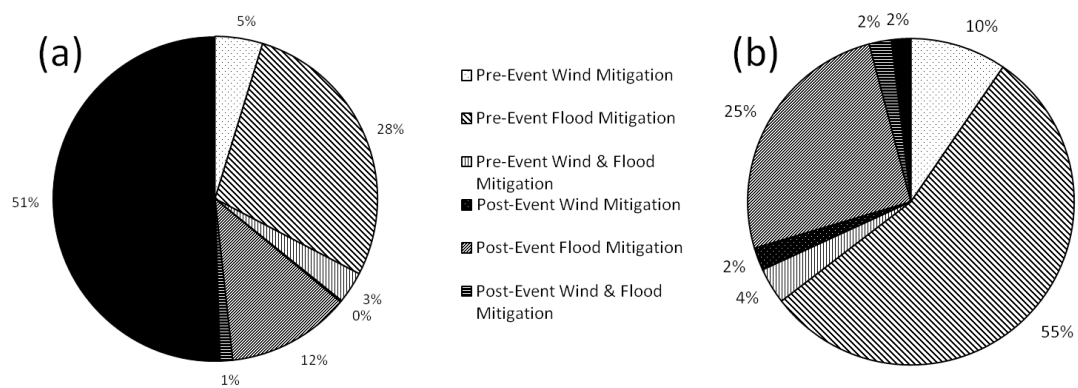


Figure 11. Distribution of total expenditures during the 30-year time horizon in terms of (a) dollars and (b) number of buildings affected.

Interestingly, mitigation funds actually spent in real life have typically focused more frequently on mitigating flood damage, rather than wind damage. For example, Rose et al. (2007) report that of 4,594 approved or completed FEMA-funded mitigation wind and flood project grants worth \$2.48 billion from 1993 to 2003, 74% of the projects and 89% of the funds were for flood rather than wind mitigation. In North Carolina in particular, the vast majority of state hurricane mitigation funds have been

spent on elevating buildings or acquiring and demolishing buildings to mitigate flood damage (Chris Crew, North Carolina State Mitigation Officer, personal communication 2009). The results of the case study analysis, thus, provide some support for the general spending patterns that have existed. Some of the specific mitigation strategies may differ, however.

#### ***6.4.2 By Timing of Investment***

Disaggregating the results by recommended timing of the investments, for an annual budget of \$8 M, Figure 11 indicates that 36% of the funds that the model recommends spending should be spent on pre-event mitigation, and 64% on post-event mitigation and acquisition. In terms of number of buildings affected, however, 69% should be modified through pre-event mitigation, and 31% should be modified through post-event mitigation or acquired and demolished after they are destroyed in a hurricane. This suggests that the best risk management strategies will include both programs to reduce losses through mitigation before an event, when they may cost more to implement but they will help avoid more losses, and programs to reduce losses through efforts made in the aftermath of a hurricane, after some damage has been done but when structural upgrades can be incorporated at reduced cost or when the option of acquiring and demolishing destroyed homes becomes feasible. Since efforts to repair and rebuild damaged houses begin quickly after a hurricane, however, plans to incorporate upgrades as part of the reconstruction effort must be in place before the

next event. Beginning to design and implement such programs after a hurricane has occurred is too late.

The model can also provide some insight into the best timing of investments at a more detailed level. For the base case analysis, Figures 12 and 13 show the pre-event and post-event recommended expenditures, respectively, year by year and by investment type. The model results indicate that funds spent on pre-event mitigation should be spent as early as possible, and then should stop when no more cost-effective pre-event mitigation alternatives remain, after Year 17 in the base case analysis. This finding is consistent with similar models for earthquakes (e.g., Dodo et al. 2005, Vaziri et al. 2010), and it makes sense since the sooner the investment in pre-event mitigation is undertaken, the longer the region can enjoy the benefits of that investment (in terms of reduced hurricane damage). Further, the model recommends pre-event spending primarily on flood mitigation first, then wind mitigation (Figure 12). This preference for flood mitigation first is the result of two factors. First, the benefit-cost metric of flood mitigations is significantly higher than wind. The average benefit-cost metric for a pre-event flood-only mitigation is 15.7, pre-event flood and wind mitigation is 6.5, and pre-event wind-only mitigation is 4. The higher benefit-cost metric for flood mitigations results in more flood mitigations chosen within the budget constraint. Second, because of unavailable data on flood mitigation in the inventory, the entire initial inventory was assumed to be in the lowest flood strength. This created an artificially high demand for flood mitigation. Similarly for post-event spending, the results suggest that post-event spending should mostly be directed toward flood

damage mitigation first, then acquisition (Figure 13). Similar conclusions can be drawn from graphs like Figures 12 and 13 but in terms of numbers of buildings affected.

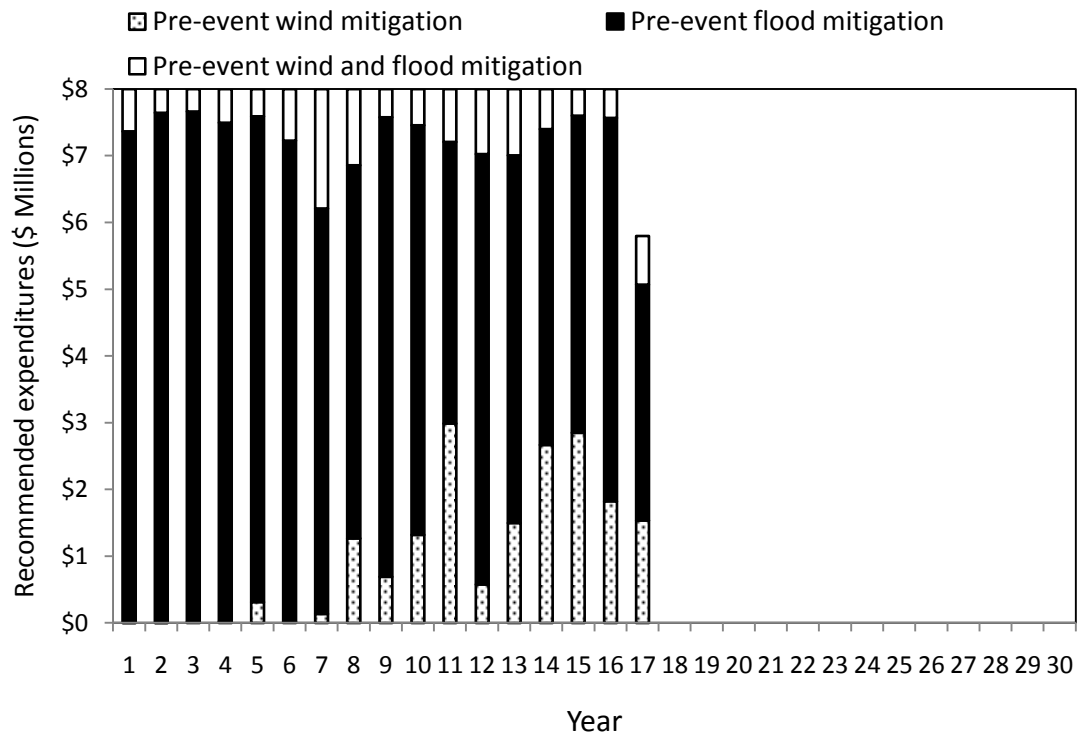


Figure 12. Recommended pre-event spending by year and type of mitigation, for base case analysis with \$8 million pre- and post-event annual budgets

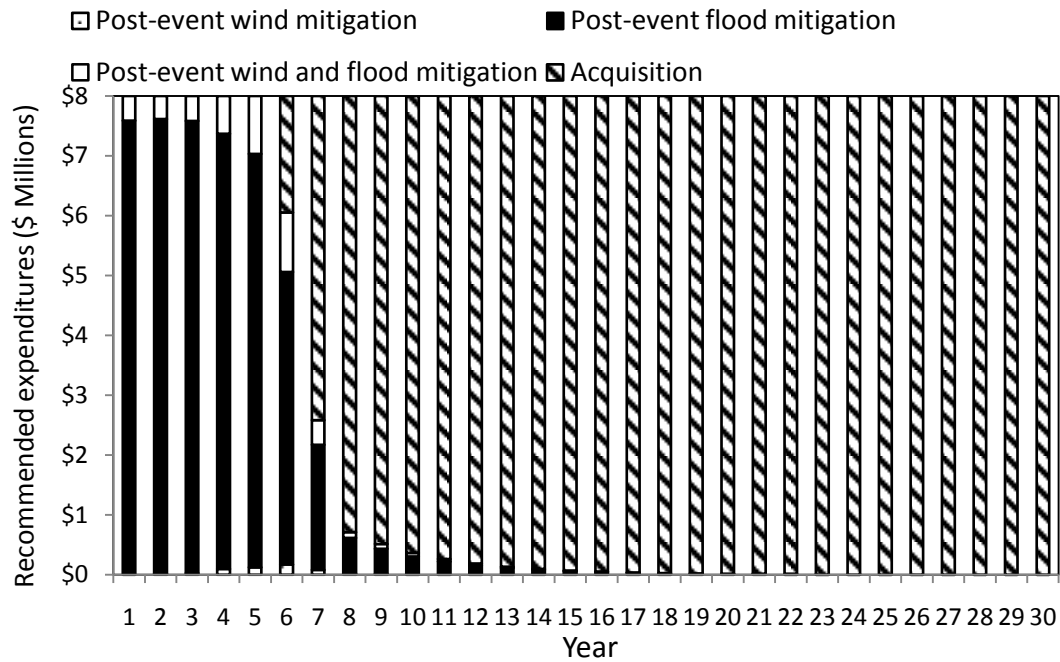


Figure 13. Recommended post-event spending by year and type of mitigation, for base case analysis with \$8 million pre- and post-event annual budgets

### 6.4.3 By Geographic Location

As the majority of mitigation spending is focused on flood, the majority of mitigation occurs in areas at risk from storm surge, which are along the coast and major waterways. Figures 14 shows the spatial distribution of buildings acquired and mitigated post-event. (The map showing numbers of building mitigated pre-event is almost the same as Figure 14.) All of the acquisitions occur in census tracts along the coast and on barrier islands. The mitigations occur along the coast and south western part of the case study area, where hazard is more severe.

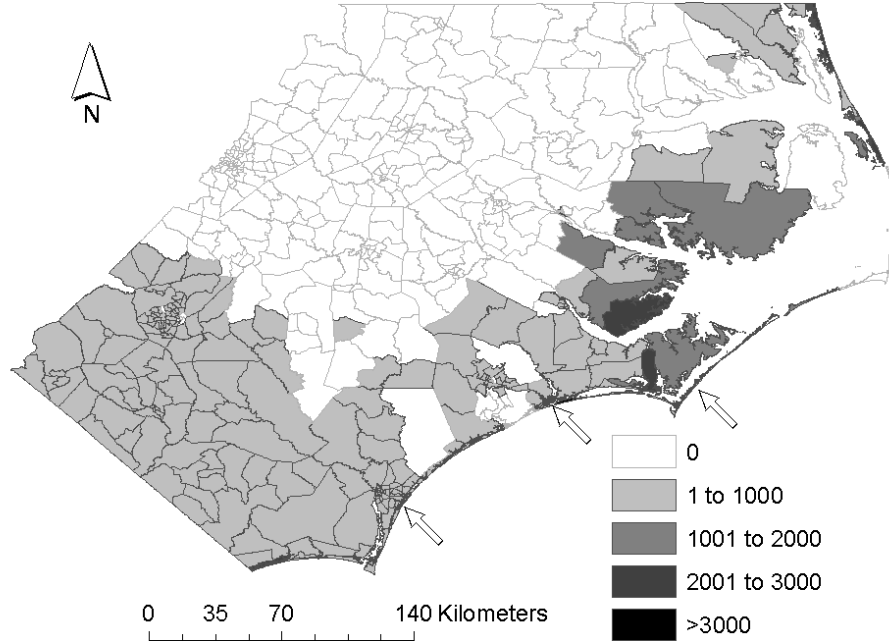


Figure 14. Map of study area indicating census tracts where buildings recommended for mitigation are located. Arrows indicate where acquisition occurs.

#### ***6.4.4 By Specific Mitigation Strategy***

Since there are many hurricane risk mitigation efforts that can be undertaken and not all are equally cost-effective, any policy should not only specify the amount to be spent on mitigation, but should indicate precisely what types of mitigations should be encouraged. A major innovation of this modeling effort is the detailed representation of the costs and benefits (i.e., effect on expected loss) of physically-realistic mitigation strategies such as, acquisition/demolition or addition of hurricane straps or shutters. As a result, the model can provide guidance on what specific types of mitigations are

recommended. The list of mitigation strategies was carefully developed to address each of the major aspects of hurricane vulnerability with a range of levels of commitment (e.g., adding nailing to roof sheathing to reduce wind damage relatively inexpensively vs. elevating an entire house to aggressively reduce severe flood damage). Realistically, the strategies also include combinations of modifications to individual components (e.g., simultaneously upgrading the roof cover, roof sheathing, and openings). Note that the model results can be interpreted as recommending the mitigation strategies as they are specifically defined in Section 5.3, or more generally as recommending any mitigation strategy with similar effect and cost. For example, a recommendation to invest in the mitigation strategy defined as changing roof cover from asphalt to high wind shingles could be interpreted more generally as a recommendation to invest in that or another strategy that would similarly increase the resistance of the roof cover at a similar cost.

For the base case analysis, with pre-event and post-event annual budgets of \$8 million, Figure 15 summarizes the resulting mitigation recommendations. Of the 68 different possible mitigations considered, 12 were chosen as a pre-event mitigation strategy at least once and 33 were chosen as a post-event mitigation strategy at least once. In both cases, however, the recommendations concentrated on a smaller set of mitigations. For 99% of the 40,800 buildings mitigated pre-event, the model recommended one of six strategies; and for 98% of the 17,900 buildings mitigated post-event, the model recommended one of eight strategies (Figure 15). (Note that all



strategies in which the only change is that appliances are elevated are grouped into *Elevate appliances* for Figure 15).

Several interesting observations can be made from Figure 15. First, both pre-event and post-event, by far the most common mitigation strategy selected is elevating appliances, a relatively minor effort to reduce flood damage. As discussed in section 6.4.2, two factors cause flood mitigation to be the most highly chosen mitigation option. First, flood mitigation, in particular elevating appliances, has a high benefit-cost metric.

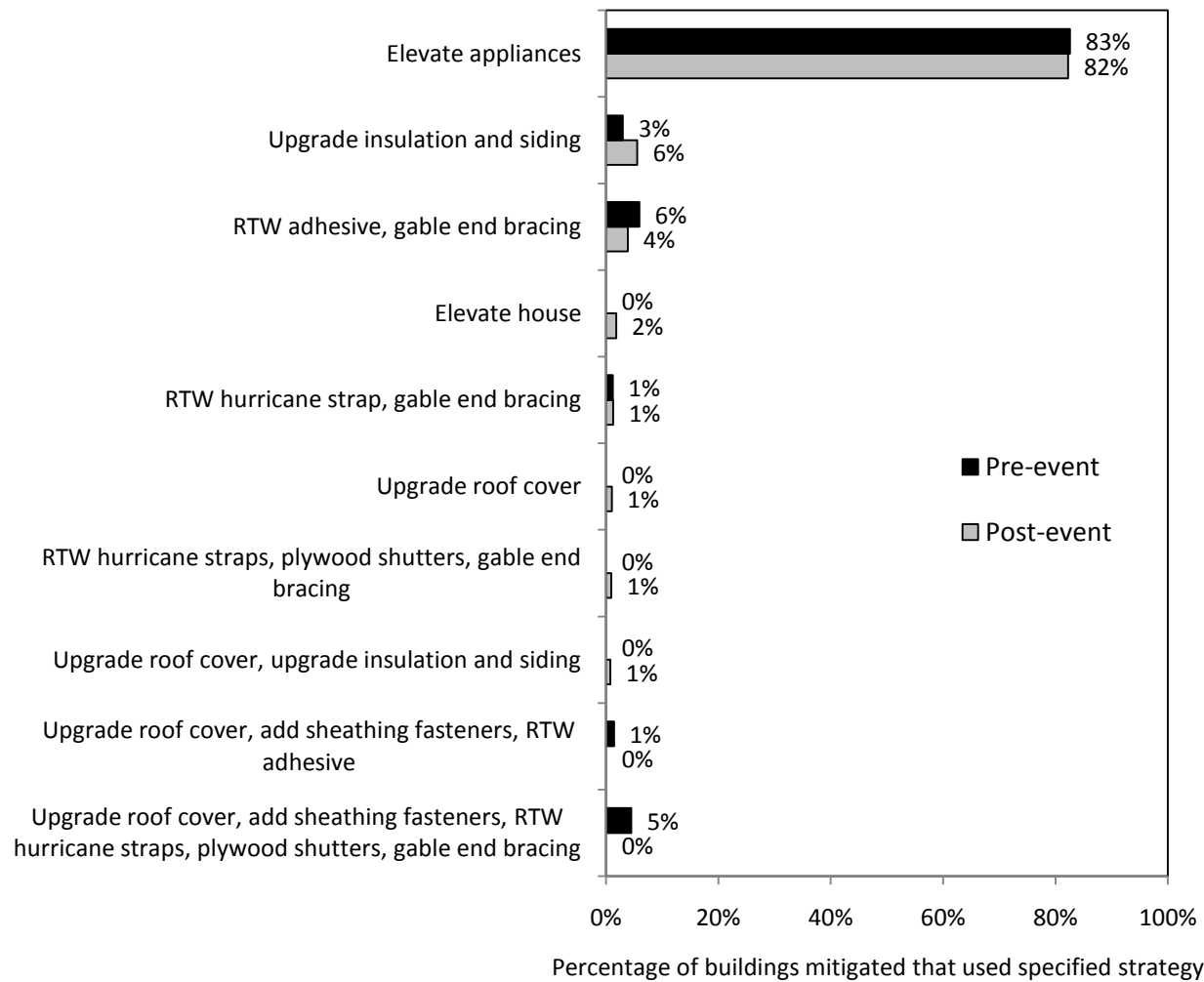


Figure 15. Percentage of buildings mitigated by the mitigation strategy used, for pre-event mitigation and post-event mitigation. (RTW refers to the roof-to-wall connection component.)

Second, the model input assumed low flood resistance for all buildings in the inventory. Elevating appliances costs between \$2,000 and \$3,000 pre-event depending on the building type, 81% of mitigation alternatives are more expensive than elevating appliances. Elevating appliances also provides a significant benefit for its low cost. The average benefit-cost metric for any mitigation strategy that includes (but is not limited to) elevating appliances is 17.4. The average benefit cost metric of all flood-only mitigations is 15.7. Also selected with some frequency was a combination of reinforcing the roof-to-wall connection (using adhesive or hurricane straps) and bracing gable ends (an approximate investment of between \$175 and \$375 pre-event), efforts designed to prevent losing the roof due to high winds (investments in roof cover are between \$3,000 and \$4,000 pre-event); and upgrading the insulation and siding (an investment between \$18,000 and \$36,000 pre-event), a moderate effort to reduce flood damage. In general, the model recommends a variety of mitigation strategies, from small to large investments, addressing wind and flood damage.

Third, the recommendations are similar for pre- and post-event mitigation, with a couple exceptions. Pre-event mitigation includes some wind mitigation combinations that were not recommended for post-event mitigation; and recommended post-event mitigations include elevating the house (an expensive flood mitigation), a wind/flood combination strategy (roof cover and insulation), and a minor wind strategy (upgrading only roof cover). Finally, for wind damage mitigation, almost all the selected strategies involve more than one building component, reflecting the reality

that a building is only as resilient as its weakest link, so it often does not make sense to upgrade only one component.

## CHAPTER 7

### CONCLUSIONS AND FUTURE WORK

#### *7.1 Conclusions*

This dissertation introduces an optimization-based regional hurricane mitigation planning model that to the best of the author's knowledge is the first in the literature. While similar optimization formulations have been developed for regional earthquake risk mitigation planning, in applying that approach to hurricanes for the first time, this new model introduces several advances. It addresses two hazards associated with hurricanes—high winds and storm surge flooding—that are correlated but have different effects on structures and require different types of strategies to mitigate. For the first time, the mitigation investment planning is based on a detailed, component-based damage assessment model, mitigation alternatives are defined in a physically realistic way at the building component level, and the investment alternative of acquiring and demolishing high risk buildings is considered. Finally, the government perspective is represented more explicitly in this model than in previous versions, considering how the government might spend funds to support mitigation. General repair and reconstruction expenditures, for example, are not assumed to come out of the budget since they are typically paid by the building owner or insurer.

The component-based approach adopted for the damage and mitigation modeling increased the already imposing computational demands of the mitigation planning

model. A second key result of this work, therefore, was developing systematic approaches to reduce the dimensions of the mitigation planning model so as to reduce computation without making simplifying assumptions that compromise the results. Specifically, approaches were developed to define sets of hurricane scenarios, building resistances, building damage states, and mitigation alternatives such that they are limited in number to reduce computation, but still meaningful. For example, rather than arbitrarily defining a few damage states that are not physically meaningful, a complete set of 2400 possible combinations of component damage states was enumerated, then carefully pared down to just 31 by removing those that are unlikely to occur in reality, and by collapsing damage states that result in similar levels of loss since that is ultimately what is important for the resource allocation optimization model. Finally, an efficient and effective heuristic solution algorithm that enables application of the regional hurricane mitigation planning for realistic situations with millions of variables was developed and tested.

The case study conducted for Eastern North Carolina offers some conclusions as well. Further investigation can examine the extent to which they are region-specific or more generally true. Case study results of course are valid for the model formulation presented (i.e., with the objectives and constraints as stated) and the input data used. Although imperfect, great care was taken to obtain the best available input data, and it is believed to be reasonable.

An increasing number of policy efforts are emerging to manage natural disaster risk at the regional level, including for example, the North Carolina Department of Insurance's program to provide insurance premium discounts for wind and hail coverage to homeowners who undertake certain mitigation efforts and mandatory earthquake retrofit policies for hospitals and schools in California. This research should help provide a systematic, transparent basis for designing and prioritizing those efforts—determining how much to spend on mitigation, how much benefit is received for that investment, and exactly how to spend it (which buildings, where, when, how).

## ***7.2 Future Work***

Engineering modeling to support regional natural disaster risk management is an emerging research area with several possible avenues of future work. As with all modeling, the results are only as good as the input data on which they rely. Efforts to improve the various types of required input data will always be helpful. In particular, data on the initial building inventory (i.e., the numbers of buildings by building category, census tract, and building resistance level) and costs of various mitigation alternatives are still relatively uncertain, although theoretically they should be obtainable. As the quality of data required by regional loss estimation models were initially quite poor but have improved over time, the hope is that the data required by regional risk management resource allocation models like the one presented here will improve as the models are more fully developed and integrated into practice.

Given the complexity and detail of the optimization model output, another avenue of future research involves developing ways to translate the model results into policy recommendations. This process involves considering how to group mitigation recommendations in way that most of the benefit is retained but the recommendations are defined simply enough that they can be implemented in practice. Translation into policy must also address any factors that are important for adoption and implementation by the different stakeholders but that have not been directly considered in the model, such as legal issues.

Finally, this model considers only the governmental perspective on regional hurricane risk management. In reality, there are multiple stakeholders—each with its own set of objectives, alternative courses of action, and constraints; and the actions of each affects the outcomes for the others. In addition to the government, insurers/reinsurers and building owners, for example, each play crucial roles in any effort to reduce regional hurricane risk. Insurers and reinsurers, for example, can be exposed to large losses in the event of a hurricane. They do not have the power to directly alter the buildings they insure, but they can provide financial incentives to encourage building owners to do so. Building owners, of course, must pay any financial losses that are not covered by insurance, and if they do not evacuate, face risk of injury or death as well. Building owners are the ones with the authority to upgrade their structures, but they often make decisions using relatively short time horizons and have limited budgets. A promising area of future research would be to extend the modeling presented herein to represent the unique perspectives of the different stakeholders explicitly. The aim



would be to develop strategies for each of the stakeholders so that when each acts in its own best interest, together they serve the needs of society as a whole.

## APPENDIX A

### MODEL BUILDINGS

The resource allocation case study examines 8 building types with the same basic design, but varying in number of stories, garage, and roof shape. The design details of the eight building types are defined in Table A.1 and shown as diagrams in Figure A.1. The approximate placement of openings is shown in Figure A.2.

Table A.1 Case study building type design

Building Type	Index	m	1	2	3	4	5	6	7	8
<b>Stories</b>	(1 = one, 2 = two)	-	1	1	1	1	2	2	2	2
<b>Geometry</b>	Length of building	L (ft)	60	60	60	60	60	60	60	60
	Width of building	W (ft)	30	30	30	30	30	30	30	30
	Mean Roof Height (d + ½e)	h (ft)	12.125	12.125	12.125	12.125	21.125	21.125	21.125	21.125
	Wall Height	d (ft)	9	9	9	9	18	18	18	18
<b>Windows</b>	Front	O <sub>wfr</sub>	3	3	3	3	6	6	6	6
	Back	O <sub>wbr</sub>	2	2	2	2	4	4	4	4
	Side regular	O <sub>wsr</sub>	2	2	2	2	4	4	4	4
	Side high pressure	O <sub>wsh</sub>	1	1	1	1	2	2	2	2
	window length (vertical)	L <sub>w</sub> (ft)	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25
	window width (horizontal)	W <sub>w</sub> (ft)	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
<b>Doors</b>	Front door	O <sub>df</sub>	1	1	1	1	1	1	1	1
	Garage door	O <sub>d<sub>g</sub></sub>	1	1	0	0	1	1	0	0
	Back doors	O <sub>db</sub>	1	1	1	1	1	1	1	1
<b>Roof</b>	Roof Shape (1 = hip, 2 = gable)	-	1	2	1	2	1	2	1	2
	Roof Pitch	r (deg)	22.62	22.62	22.62	22.62	22.62	22.62	22.62	22.62
	Roof Pitch	r (rise/run)	5/12	5/12	5/12	5/12	5/12	5/12	5/12	5/12
	Roof height	e (ft)	6.25	6.25	6.25	6.25	6.25	6.25	6.25	6.25
	roof width	c (ft)	16.25	16.25	16.25	16.25	16.25	16.25	16.25	16.25
	hip ridge length	a (ft)	22.11	22.11	22.11	22.11	22.11	22.11	22.11	22.11
	hip ridge angle	θ (deg)	47.29	47.29	47.29	47.29	47.29	47.29	47.29	47.29
	truss spacing	t (ft)	2	2	2	2	2	2	2	2
<b>Sheathing Panel</b>	Length	L <sub>p</sub> (ft)	8	8	8	8	8	8	8	8
	Width	W <sub>p</sub> (ft)	4	4	4	4	4	4	4	4

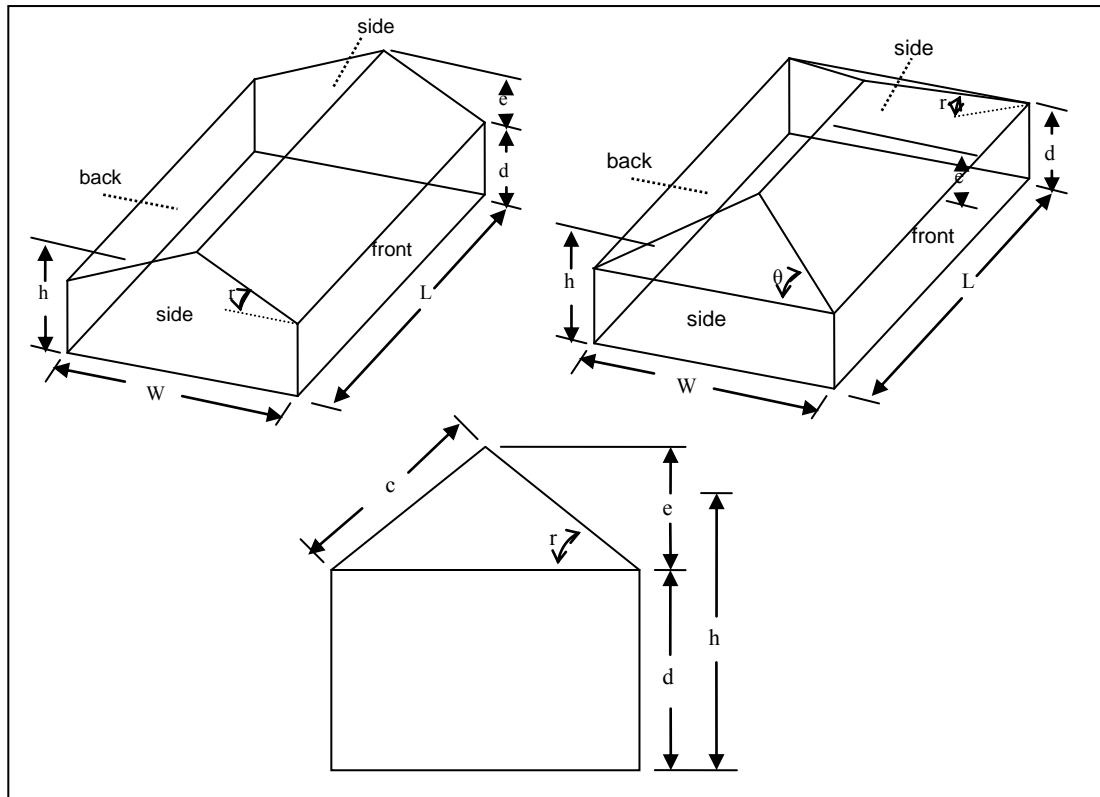


Figure A.1 Model building dimensions

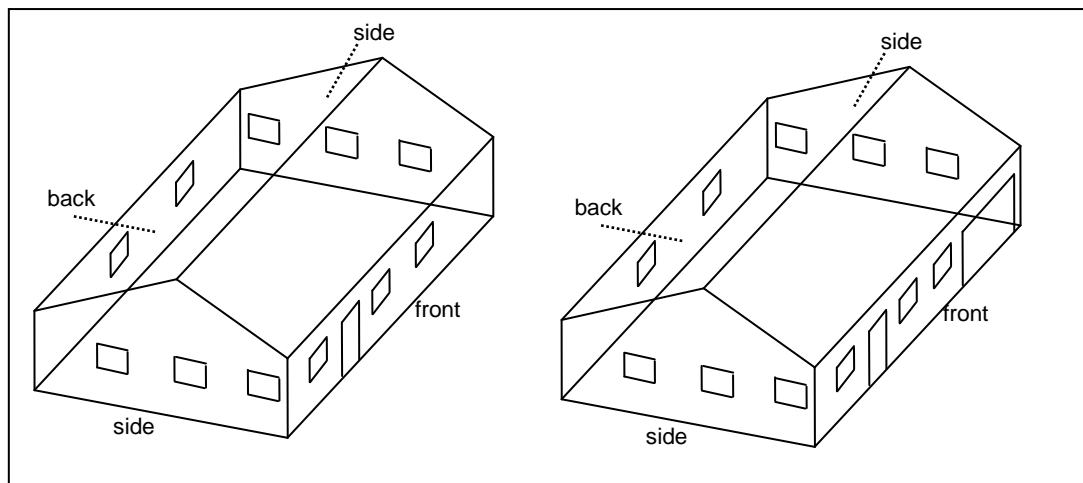


Figure A.2 Model building opening locations

All buildings have a 60 ft x 30 ft footprint, roof pitch of 5'/12' (22.62°), one front door and one back door, roof trusses spaced at 2 ft on center, and roof sheathing panels of dimension 4 ft x 8ft, and windows of standard dimension 3.5 ft x 2.25 ft. One story buildings have 3 front windows, 2 back windows, and 3 side windows. The side windows are divided between those in the regular pressure area and those in the high pressure area (for details on the load calculation see Appendix D). Two story buildings have twice the number of windows, divided equally in the pressure areas.

## APPENDIX B

### BUILDING RESISTANCES AND MITIGATION ALTERNATIVES

The hurricane wind and flood resistance of each building is determined by the combination of resistances of each of the building's six structural components. To evaluate real mitigation strategies, construction alternatives were identified for each of the six building components. The alternative configurations for each of the components is described in Table B.1.

As described in Chapter 5, with two alternatives for roof cover, three for roof covering, four for openings, three for roof-to-wall connections, two for walls, and four for flood, there are theoretically 576 possible building resistances *c*. With the goal of defining a set of building resistances such that each is physically realistic and reasonably common, they are substantially different from one another, and together they are relatively few in number while representing the range of possible resistances, the set of 576 building resistances was reduced to 68. The number of building resistances and the number of possible mitigation options was reduced using the following 9 rules to eliminate alternatives.

Table B.1 Component configurations

Roof Cover

RC 1	Asphalt Shingles
RC 2	High Wind Shingles

Roof Sheathing

RS 1	Panels fastened with 8d nails @ 6" o.c. edge and 12" o.c. interior
RS 2	Panels fastened with 8d nails @ 3" o.c. edge and 6" o.c. interior
RS 3	Panels fastened with 8d nails @ 6" o.c. edge and 12" o.c. interior and construction adhesive applied to lumber-to-sheathing joint

Roof to Wall Connections

RWC 1	Toe nail connections @ 24" o.c.
RWC 2	Toe nail connection & structural adhesive
RWC 3	Toe nail connection & hurricane strap @ 24" o.c.

Openings

O 1	Standard Windows, Doors, Garage Door
O 2	Impact Resistant Windows, Hurricane Doors, Hurricane Garage Door
O 3	Plywood shutters on Windows, Doors, Garage Door
O 4	Hurricane Shutters on Windows, Doors, Garage Door

Wall

W1	No Gable End Bracing
W2	Gable End Bracing

Flood Vulnerability

F 1	No Flood Protection
F 2	Elevated Appliances, Upgraded Electrical
F 3	Upgraded Insulation and Siding
F 4	Elevated Structure

Rule 1: Mitigate roof cover only when sheathing is also mitigated (or already strong).

Replacing roof cover would be one of the last mitigation choices. Strong roof sheathing would be a prerequisite, as strong roof cover will do very little to reinforce a roof without strong roof sheathing. Therefore, roof cover is only mitigated in two cases: (a) when roof sheathing is removed and replaced because the roof cover would have to be replaced anyway and (b) when roof sheathing is already strong. This

additionally means that many component resistance combinations with strong roof sheathing and poor roof cover can be eliminated.

Rule 2: Adhesive would be applied to both roof sheathing and roof-to-wall connections at the same time, not individually.

Strengthening roof sheathing with adhesive can be done without removing and replacing roof sheathing or shingles. If a home owner wanted to strengthen his roof sheathing with adhesive, it would make sense at that time to also strengthen his roof-to-wall connections with adhesive. Therefore, mitigation with structural adhesive involves both roof sheathing and roof-to-wall connections. This additionally means that many component resistance combinations with adhesive on one element but not the other can be eliminated.

Rule 3: Hurricane straps will only be added in conjunction with upgrading roof sheathing or installing gable end bracing.

It is expensive to add hurricane straps to roof-to-wall connections without removing a roof. It is likely, therefore, that roof-to-wall connections would be strengthened with hurricane straps when roof sheathing is being removed and replaced. Therefore, the mitigation of adding hurricane straps will only be considered when some other work is being done to the roof, such as upgrading roof sheathing or installing gable end bracing.

Rule 4: When mitigating openings, all openings are upgraded.

When mitigating openings, it is assumed that all openings will be mitigated at the same time. While it may occur in reality that openings would be upgraded over a period of time, the building envelope will not benefit from the mitigation until all openings are upgraded. Additionally, taking into account the mitigation of some openings over others adds unnecessary complexity to the program.

Rule 5: Openings are mitigated by either strengthening the opening or adding shutters, not both.

Openings can be mitigated through shutters (plywood or hurricane grade) or by upgrading the glass in the windows to heat strengthened glass. Financially, a homeowner would choose to either add shutters or strengthen the glass, but not to do both at the same time. It would be possible after upgrading to heat strengthened glass to add hurricane shutters as a mitigation measure, however this would be a last mitigation measure after the roof had been strengthened, as shutters would be a redundancy and the roof might still be vulnerable. Therefore, the opening mitigations recognized are either the upgrade to strengthened glass OR the upgrade to shutters.

Rule 6: Gable end bracing will occur in conjunction with any roof mitigation.

When any kind of roof sheathing or roof-to-wall connection mitigation occurs, the gable end walls of gable roofs should also be reinforced. This is a simple mitigation



measure and is relatively easy to perform when already mitigating any portion of the roof.

Rule 7: The weakest component will be mitigated first before upgrading moderately strong elements.

If a home has both poor openings and a poor roof, then mitigating one or the other will not significantly improve the overall strength of the home. In order to reduce the vulnerability of the entire building envelope, both the roof system and the openings must be strengthened. Therefore the mitigation alternatives chosen as most likely, and considered in this model, are those where both the roof and the openings are improved. Likewise, if a home has weak openings but a strong roof, or vice versa, then mitigating the strong component would not improve the strength of the overall building envelope. Rather, the more likely mitigation would be to strengthen the weak component. Once the openings and roof are comparably strong, then either could be mitigated further. Additionally, this means that many component resistance combinations with combinations of extremely strong and extremely weak elements can be eliminated as highly unlikely.

Rule 8: Flood vulnerability options are mutually exclusive.

Rule 9: A structure with weak wind protection is likely to have weak flood protection.

If a home has both weak openings and a weak roof, or extremely weak of either, then it is likely to have no or very little flood protection. Elevating the structure without other mitigations would not make sense, as the building would still be extremely vulnerable to wind damage. The only flood mitigation that would make sense would be elevating appliances or upgrading insulation and siding. Additionally, this means that many component resistance combinations with weak wind protection and strong flood protection can be eliminated as unlikely.

Based on the 9 rules defined above, a summary of the building resistances and mitigation alternatives is displayed in diagram form in Figure 4, Chapter 5. A more detailed diagram is presented below in four parts, Figures B.1, B.2, B.3, and B.4. Each figure presents a portion of the building resistances and the various mitigation alternatives between the resistances. Each resistance contains the component resistance values as described in Table B.1.

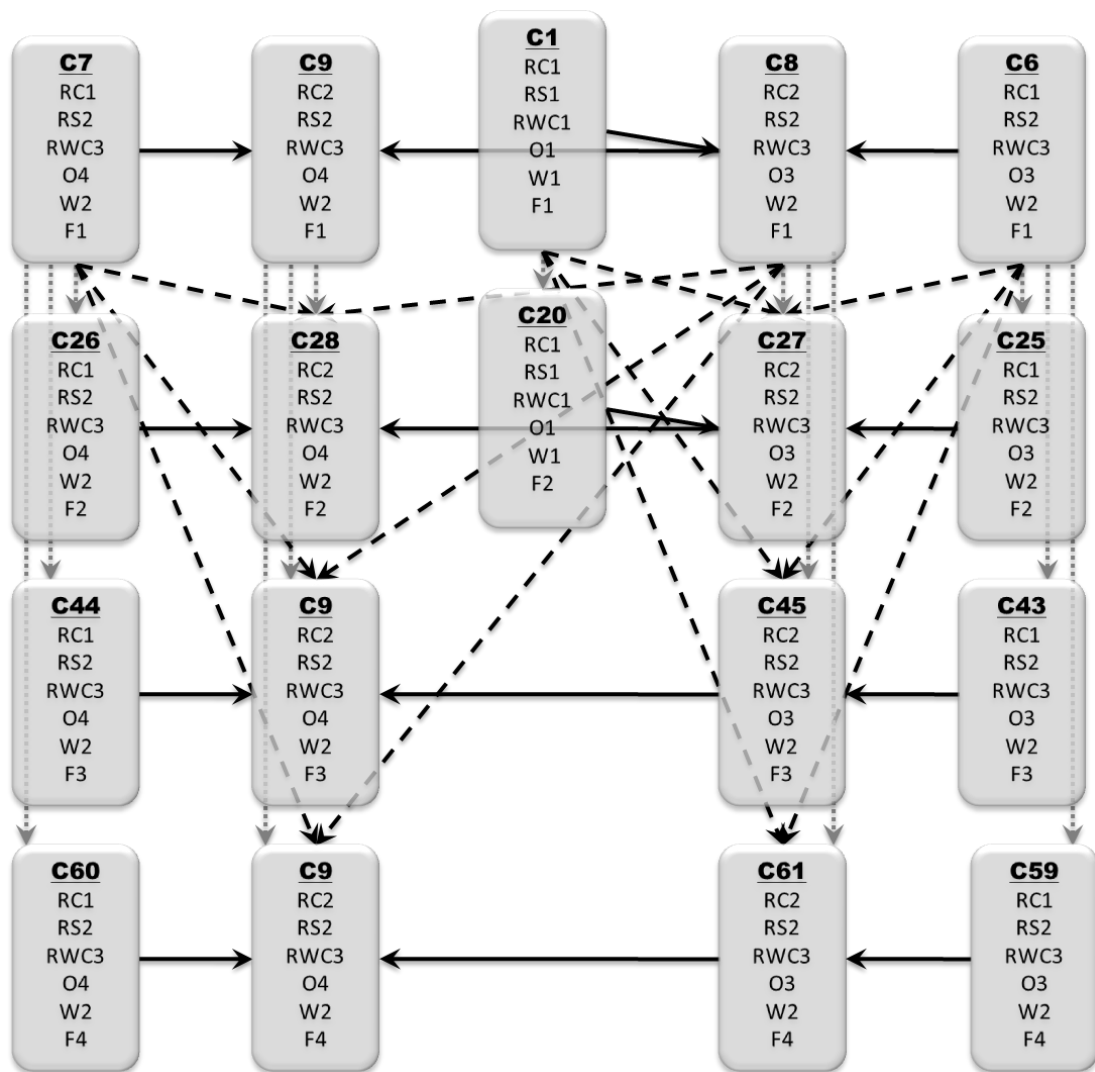


Figure B.1. Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study.

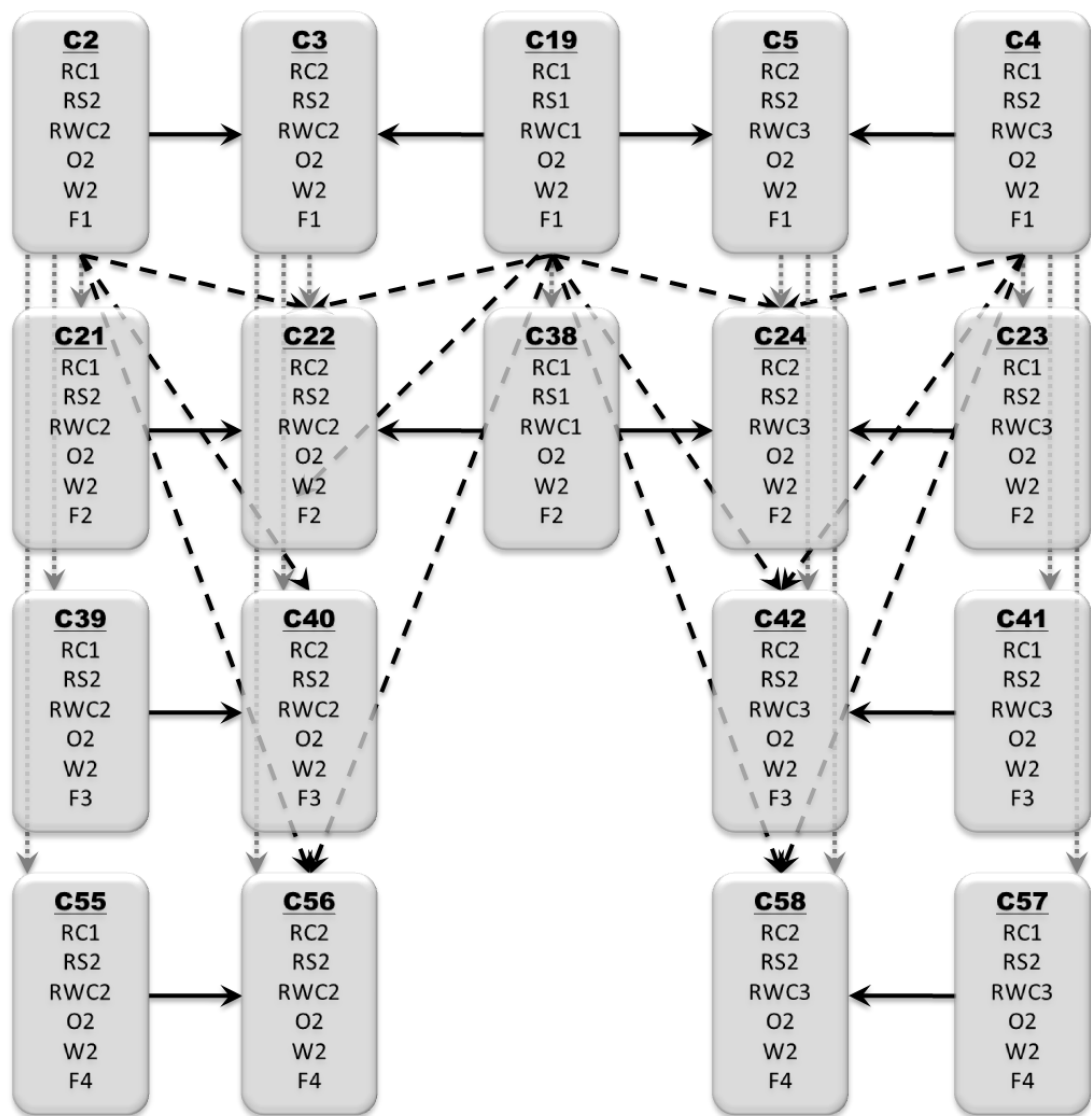


Figure B.2. Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study.

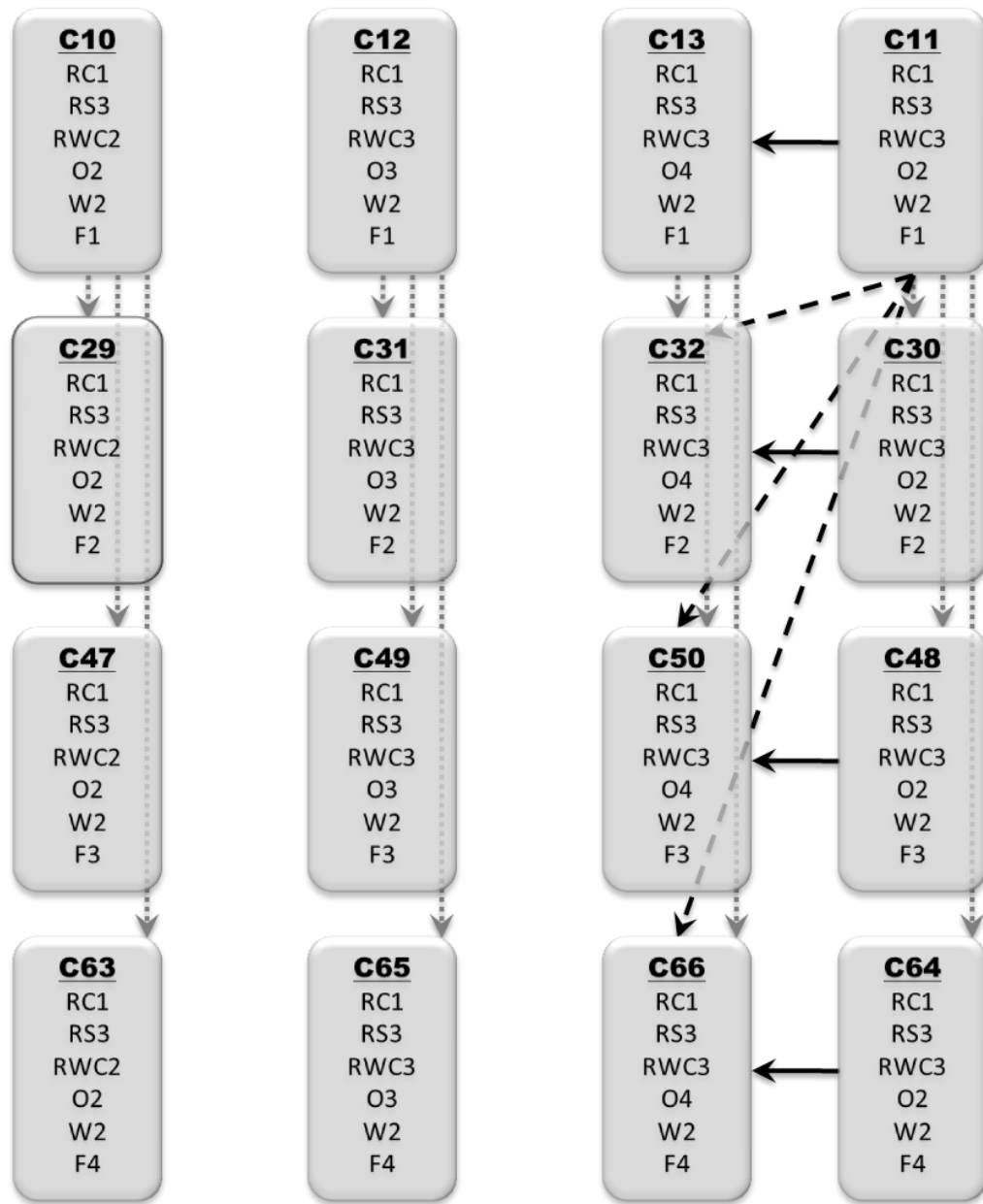


Figure B.3. Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study.

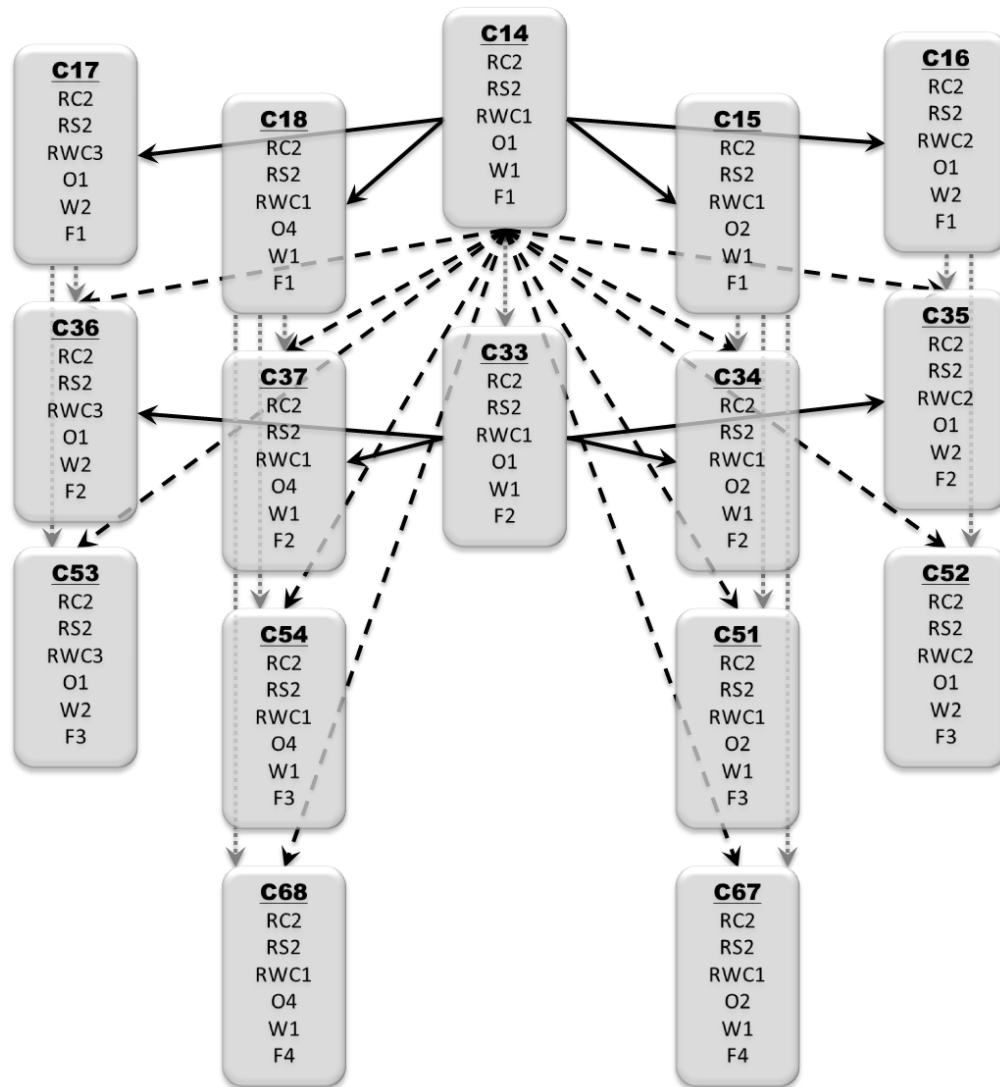


Figure B.4. Detailed diagram of a portion of component resistance combinations and mitigation alternatives in the case study.

In Figures B.1, B.2, B.3, and B.4, each rectangle represents a building resistance. Within each rectangle are the individual component resistances, the full description of which can be found in Table B.1. Each arrow represents a mitigation alternative, changing from one building resistance to another. A dotted gray arrow represents a mitigation alternative to reduce flood damage, a solid black arrow represents a wind mitigation alternative, and a dotted black arrow represents an alternative mitigating for both flood and wind.

## APPENDIX C

### BUILDING DAMAGE STATE DEFINITIONS

As described in Chapter 5, the optimization model quantifies damage in terms of the component damage states described in Table 3. With four damage states defined for roof cover, five for roof sheathing, two for roof-to-wall connections, five for openings, two for walls, and six for flood, considering all possible combinations would result in 2400 possible building damage states. As with building resistances, that set was drastically reduced to 31 by removing those that are unlikely to occur in reality, and by collapsing damage states that result in similar levels of loss since that is ultimately what is important for the resource allocation optimization model.

The final set of damage states was obtained by first examining the probability of wind damage state. Excluding the six flood damage states, there are 400 possible wind related damage states. Of those damage states, 300 include wall and/or roof-to-wall connection damage that would require complete reconstruction of the building. Any damage state resulting in complete reconstruction of the building can be collapsed into the maximum “totaled” damage state, DS 31.

Next, the remaining 100 damages states were examined for impossibility. Damage states with a combination of high roof sheathing damage and low roof cover damage are impossible in reality, and do not occur in the loss model. Because of the physical

connection between roof sheathing and roof cover, roof cover will always have at least as much damage as roof sheathing. Therefore, damage state combinations with roof sheathing in damage state 4 and roof cover in damage state 0, 1, or 2 were eliminated and damage state combinations with roof sheathing in damage state 3 and roof cover in damage state 0, 1 were eliminated. This removed 25 additional damage states.

Further examination of the roof sheathing damage states revealed that roof sheathing damage state 1 can be collapsed into roof sheathing damage state 0. Roof sheathing damage state 1 is defined as damage of less than 2% of the entire roof. This is effectively negligible and the cost to repair this damage is estimated as \$0. Therefore the cost to repair the roof sheathing in damage state 1 is equal to the cost to repair the roof sheathing in damage state 0. These two damage states are effectively the same and any building that experiences a damage state combination with roof sheathing in damage state 1 was considered to be experiencing roof sheathing in damage state 0. This removed 19 additional damage states.

The above reductions resulted in 56 probable wind damage states, which were then combined with the six possible flood damage states to obtain a list of 336 possible damage states. Damage states with flood damage state 5 were considered to be in the maximum “totaled” damage state and were collapsed into DS 31. This reduced the number of damage states to 278.

Reconstruction costs,  $R$ , for each of the 278 damage states were calculated and compared in an effort to further reduce the number of damage states. Table C.1



displays the reconstruction cost estimates for the 278 damage states for a Type 1 Building. Any damage state resulting in over 50% damage is considered a “totaled” damage state because the building will require complete reconstruction. Using this rule, 81 additional damage states were collapsed into the “totaled” damage state, DS 31.

Finally, the remaining 197 damage states were combined into groups based on their approximate reconstruction cost,  $R$ . For example, any damage requiring less than \$1,000 to repair was grouped into DS 2. Damage requiring  $> \$1,000$  and  $< \$2,000$  was grouped into DS 3. This process grouped damage states with similar reconstruction costs into the groups shown in Table C.1.

Table C.1 Damage state grouping by reconstruction cost

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
1	\$ -	0	0	0	0	0	0	\$ -
2	\$ 636.90	0	1	0	0	0	0	\$ 424.60
		0	2	0	0	0	0	\$ 849.20
3	\$ 1,593.74	1	0	0	0	0	0	\$ 1,089.22
		1	1	0	0	0	0	\$ 1,513.82
		0	3	0	0	0	0	\$ 1,833.50
		1	2	0	0	0	0	\$ 1,938.42
4	\$ 3,931.76	1	3	0	0	0	0	\$ 2,922.72
		0	4	0	0	0	0	\$ 4,940.80
5	\$ 7,002.31	1	4	0	0	0	0	\$ 6,030.02
		0	0	0	0	0	1	\$ 7,276.15
		0	1	0	0	0	1	\$ 7,700.75
6	\$ 8,967.30	0	2	0	0	0	1	\$ 8,125.35
		1	0	0	0	0	1	\$ 8,365.37
		1	1	0	0	0	1	\$ 8,789.97
		0	3	0	0	0	1	\$ 9,109.65
		1	2	0	0	0	1	\$ 9,214.57
		1	3	0	0	0	1	\$ 10,198.87
7	\$ 13,101.58	0	4	0	0	0	1	\$ 12,216.95
		0	0	2	0	0	0	\$ 13,229.29
		1	4	0	0	0	1	\$ 13,306.17
		0	1	2	0	0	0	\$ 13,653.89

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
8	\$ 14,920.44	0	2	2	0	0	0	\$ 14,078.49
		1	0	2	0	0	0	\$ 14,318.52
		1	1	2	0	0	0	\$ 14,743.12
		0	3	2	0	0	0	\$ 15,062.79
		1	2	2	0	0	0	\$ 15,167.72
		1	3	2	0	0	0	\$ 16,152.02
9	\$ 18,714.70	0	4	2	0	0	0	\$ 18,170.09
		1	4	2	0	0	0	\$ 19,259.32
10	\$ 21,242.65	0	0	2	0	0	1	\$ 20,505.44
		0	1	2	0	0	1	\$ 20,930.04
		0	2	2	0	0	1	\$ 21,354.64
		1	0	2	0	0	1	\$ 21,594.66
		0	0	0	0	0	2	\$ 21,828.45
11	\$ 22,884.97	1	1	2	0	0	1	\$ 22,019.26
		0	1	0	0	0	2	\$ 22,253.05
		0	3	2	0	0	1	\$ 22,338.94
		1	2	2	0	0	1	\$ 22,443.86
		0	2	0	0	0	2	\$ 22,677.65
		1	0	0	0	0	2	\$ 22,917.67
		1	1	0	0	0	2	\$ 23,342.27
		1	3	2	0	0	1	\$ 23,428.16
		0	3	0	0	0	2	\$ 23,661.95
		1	2	0	0	0	2	\$ 23,766.87

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
12	\$ 25,875.53	1	3	0	0	0	2	\$ 24,751.17
		0	4	2	0	0	1	\$ 25,446.24
		1	4	2	0	0	1	\$ 26,535.46
		0	4	0	0	0	2	\$ 26,769.25
13	\$ 28,543.06	1	4	0	0	0	2	\$ 27,858.47
		2	0	0	0	0	0	\$ 27,937.39
		2	1	0	0	0	0	\$ 28,361.99
		2	2	0	0	0	0	\$ 28,786.59
		2	3	0	0	0	0	\$ 29,770.89
14	\$ 32,878.19	2	4	0	0	0	0	\$ 32,878.19
15	\$ 36,096.12	0	0	2	0	0	2	\$ 35,057.74
		2	0	0	0	0	1	\$ 35,213.54
		0	1	2	0	0	2	\$ 35,482.34
		2	1	0	0	0	1	\$ 35,638.14
		0	2	2	0	0	2	\$ 35,906.94
		2	2	0	0	0	1	\$ 36,062.74
		1	0	2	0	0	2	\$ 36,146.96
		0	0	0	0	0	3	\$ 36,380.75
		1	1	2	0	0	2	\$ 36,571.56
		0	1	0	0	0	3	\$ 36,805.35
		0	3	2	0	0	2	\$ 36,891.24
		1	2	2	0	0	2	\$ 36,996.16

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
16	\$ 37,736.48	2	3	0	0	0	1	\$ 37,047.04
		0	2	0	0	0	3	\$ 37,229.95
		1	0	0	0	0	3	\$ 37,469.97
		1	1	0	0	0	3	\$ 37,894.57
		1	3	2	0	0	2	\$ 37,980.46
		0	3	0	0	0	3	\$ 38,214.25
		1	2	0	0	0	3	\$ 38,319.17
17	\$ 39,818.78	1	3	0	0	0	3	\$ 39,303.47
		0	4	2	0	0	2	\$ 39,998.54
		2	4	0	0	0	1	\$ 40,154.34
18	\$ 41,799.16	1	4	2	0	0	2	\$ 41,087.76
		2	0	2	0	0	0	\$ 41,166.68
		0	4	0	0	0	3	\$ 41,321.55
		2	1	2	0	0	0	\$ 41,591.28
		2	2	2	0	0	0	\$ 42,015.88
		1	4	0	0	0	3	\$ 42,410.77
		2	3	2	0	0	0	\$ 43,000.18
19	\$ 46,107.48	2	4	2	0	0	0	\$ 46,107.48

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
20	\$ 49,163.12	2	0	2	0	0	1	\$ 48,442.83
		3	0	0	0	0	0	\$ 48,659.86
		3	0	2	0	0	0	\$ 48,762.98
		2	1	2	0	0	1	\$ 48,867.43
		3	1	0	0	0	0	\$ 49,084.46
		3	1	2	0	0	0	\$ 49,187.58
		2	2	2	0	0	1	\$ 49,292.03
		3	2	0	0	0	0	\$ 49,509.06
		0	0	2	0	0	3	\$ 49,610.04
		3	2	2	0	0	0	\$ 49,612.18
		2	0	0	0	0	2	\$ 49,765.83

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
21	\$ 50,978.11	0	1	2	0	0	3	\$ 50,034.64
		2	1	0	0	0	2	\$ 50,190.43
		2	3	2	0	0	1	\$ 50,276.33
		0	2	2	0	0	3	\$ 50,459.24
		3	3	0	0	0	0	\$ 50,493.36
		3	3	2	0	0	0	\$ 50,596.48
		2	2	0	0	0	2	\$ 50,615.03
		1	0	2	0	0	3	\$ 50,699.26
		2	0	3	0	0	0	\$ 50,722.26
		3	0	3	0	0	0	\$ 50,722.26
		3	0	4	0	0	0	\$ 50,722.26
		0	0	0	0	0	4	\$ 50,933.04
		1	1	2	0	0	3	\$ 51,123.86
		2	1	3	0	0	0	\$ 51,146.86
		3	1	3	0	0	0	\$ 51,146.86
		3	1	4	0	0	0	\$ 51,146.86
		0	1	0	0	0	4	\$ 51,357.64
		0	3	2	0	0	3	\$ 51,443.54
		1	2	2	0	0	3	\$ 51,548.46
		2	2	3	0	0	0	\$ 51,571.46
		3	2	3	0	0	0	\$ 51,571.46
		3	2	4	0	0	0	\$ 51,571.46
		2	3	0	0	0	2	\$ 51,599.33
		0	2	0	0	0	4	\$ 51,782.24



Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
22	\$ 52,904.25	1	0	0	0	0	4	\$ 52,022.27
		1	1	0	0	0	4	\$ 52,446.87
		1	3	2	0	0	3	\$ 52,532.76
		2	3	3	0	0	0	\$ 52,555.76
		3	3	3	0	0	0	\$ 52,555.76
		3	3	4	0	0	0	\$ 52,555.76
		0	3	0	0	0	4	\$ 52,766.54
		1	2	0	0	0	4	\$ 52,871.47
		2	4	2	0	0	1	\$ 53,383.63
		3	4	0	0	0	0	\$ 53,600.66
		3	4	2	0	0	0	\$ 53,703.78
23	\$ 55,462.07	1	3	0	0	0	4	\$ 53,855.77
		0	4	2	0	0	3	\$ 54,550.84
		2	4	0	0	0	2	\$ 54,706.63
		1	4	2	0	0	3	\$ 55,640.06
		2	4	3	0	0	0	\$ 55,663.06
		3	4	3	0	0	0	\$ 55,663.06
		3	4	4	0	0	0	\$ 55,663.06
		0	4	0	0	0	4	\$ 55,873.84
		3	0	0	0	0	1	\$ 55,936.01

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
24	\$ 57,194.31	3	0	2	0	0	1	\$ 56,039.13
		3	1	0	0	0	1	\$ 56,360.61
		3	1	2	0	0	1	\$ 56,463.73
		3	2	0	0	0	1	\$ 56,785.21
		3	2	2	0	0	1	\$ 56,888.33
		1	4	0	0	0	4	\$ 56,963.07
		3	3	0	0	0	1	\$ 57,769.51
		3	3	2	0	0	1	\$ 57,872.63
		2	0	3	0	0	1	\$ 57,998.41
		3	0	3	0	0	1	\$ 57,998.41
		3	0	4	0	0	1	\$ 57,998.41
25	\$ 59,034.17	2	1	3	0	0	1	\$ 58,423.01
		3	1	3	0	0	1	\$ 58,423.01
		3	1	4	0	0	1	\$ 58,423.01
		2	2	3	0	0	1	\$ 58,847.61
		3	2	3	0	0	1	\$ 58,847.61
		3	2	4	0	0	1	\$ 58,847.61
		2	3	3	0	0	1	\$ 59,831.91
		3	3	3	0	0	1	\$ 59,831.91
		3	3	4	0	0	1	\$ 59,831.91

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
26	\$ 62,278.25	3	4	0	0	0	1	\$ 60,876.81
		3	4	2	0	0	1	\$ 60,979.93
		2	4	3	0	0	1	\$ 62,939.21
		3	4	3	0	0	1	\$ 62,939.21
		3	4	4	0	0	1	\$ 62,939.21
		2	0	2	0	0	2	\$ 62,995.13
27	\$ 64,750.44	2	1	2	0	0	2	\$ 63,419.73
		2	2	2	0	0	2	\$ 63,844.33
		0	0	2	0	0	4	\$ 64,162.34
		2	0	0	0	0	3	\$ 64,318.13
		0	1	2	0	0	4	\$ 64,586.94
		2	1	0	0	0	3	\$ 64,742.73
		2	3	2	0	0	2	\$ 64,828.63
		0	2	2	0	0	4	\$ 65,011.54
		2	2	0	0	0	3	\$ 65,167.33
		1	0	2	0	0	4	\$ 65,251.56
		1	1	2	0	0	4	\$ 65,676.16
		0	3	2	0	0	4	\$ 65,995.84
28	\$ 66,818.34	1	2	2	0	0	4	\$ 66,100.76
		2	3	0	0	0	3	\$ 66,151.63
		1	3	2	0	0	4	\$ 67,085.06
		2	4	2	0	0	2	\$ 67,935.93

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
29	\$ 70,091.18	0	4	2	0	0	4	\$ 69,103.14
		2	4	0	0	0	3	\$ 69,258.93
		1	4	2	0	0	4	\$ 70,192.36
		3	0	0	0	0	2	\$ 70,488.31
		3	0	2	0	0	2	\$ 70,591.43
		3	1	0	0	0	2	\$ 70,912.91
30	\$ 72,024.13	3	1	2	0	0	2	\$ 71,016.03
		3	2	0	0	0	2	\$ 71,337.51
		3	2	2	0	0	2	\$ 71,440.63
		3	3	0	0	0	2	\$ 72,321.81
		3	3	2	0	0	2	\$ 72,424.93
		2	0	3	0	0	2	\$ 72,550.70
		3	0	3	0	0	2	\$ 72,550.70
		3	0	4	0	0	2	\$ 72,550.70
31	\$ 145,522.98	2	1	3	0	0	2	\$ 145,522.98
		3	1	3	0	0	2	\$ 145,522.98
		3	1	4	0	0	2	\$ 145,522.98
		2	2	3	0	0	2	\$ 145,522.98
		3	2	3	0	0	2	\$ 145,522.98
		3	2	4	0	0	2	\$ 145,522.98
		2	3	3	0	0	2	\$ 145,522.98
		3	3	3	0	0	2	\$ 145,522.98

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
		3	3	4	0	0	2	\$ 145,522.98
		3	4	0	0	0	2	\$ 145,522.98
		3	4	2	0	0	2	\$ 145,522.98
		2	4	3	0	0	2	\$ 145,522.98
		3	4	3	0	0	2	\$ 145,522.98
		3	4	4	0	0	2	\$ 145,522.98
		2	0	2	0	0	3	\$ 145,522.98
		2	1	2	0	0	3	\$ 145,522.98
		2	2	2	0	0	3	\$ 145,522.98
		2	0	0	0	0	4	\$ 145,522.98
		2	1	0	0	0	4	\$ 145,522.98
		2	3	2	0	0	3	\$ 145,522.98
		2	2	0	0	0	4	\$ 145,522.98
		2	3	0	0	0	4	\$ 145,522.98
		2	4	2	0	0	3	\$ 145,522.98
		2	4	0	0	0	4	\$ 145,522.98
		3	0	0	0	0	3	\$ 145,522.98
		3	0	2	0	0	3	\$ 145,522.98
		3	1	0	0	0	3	\$ 145,522.98
		3	1	2	0	0	3	\$ 145,522.98
		3	2	0	0	0	3	\$ 145,522.98
		3	2	2	0	0	3	\$ 145,522.98
		3	3	0	0	0	3	\$ 145,522.98
		3	3	2	0	0	3	\$ 145,522.98

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
		2	0	3	0	0	3	\$ 145,522.98
		3	0	3	0	0	3	\$ 145,522.98
		3	0	4	0	0	3	\$ 145,522.98
		2	1	3	0	0	3	\$ 145,522.98
		3	1	3	0	0	3	\$ 145,522.98
		3	1	4	0	0	3	\$ 145,522.98
		2	2	3	0	0	3	\$ 145,522.98
		3	2	3	0	0	3	\$ 145,522.98
		3	2	4	0	0	3	\$ 145,522.98
		2	3	3	0	0	3	\$ 145,522.98
		2	3	4	0	0	3	\$ 145,522.98
		3	3	3	0	0	3	\$ 145,522.98
		3	3	4	0	0	3	\$ 145,522.98
		3	4	0	0	0	3	\$ 145,522.98
		3	4	2	0	0	3	\$ 145,522.98
		2	4	3	0	0	3	\$ 145,522.98
		3	4	3	0	0	3	\$ 145,522.98
		3	4	4	0	0	3	\$ 145,522.98
		2	0	2	0	0	4	\$ 145,522.98
		2	1	2	0	0	4	\$ 145,522.98
		2	2	2	0	0	4	\$ 145,522.98
		2	3	2	0	0	4	\$ 145,522.98
		2	4	2	0	0	4	\$ 145,522.98
		3	0	0	0	0	4	\$ 145,522.98
		3	0	2	0	0	4	\$ 145,522.98

Table C.1 (Continued)

Building Damage State	Average Cost	Component Damage States						Reconstruction Costs
		Roof Cover	Openings	Roof Sheathing	Roof-to-Wall Connections	Walls	Flood	
		3	1	0	0	0	4	\$ 145,522.98
		3	1	2	0	0	4	\$ 145,522.98
		3	2	0	0	0	4	\$ 145,522.98
		3	2	2	0	0	4	\$ 145,522.98
		3	3	0	0	0	4	\$ 145,522.98
		3	3	2	0	0	4	\$ 145,522.98
		2	0	3	0	0	4	\$ 145,522.98
		3	0	3	0	0	4	\$ 145,522.98
		3	0	4	0	0	4	\$ 145,522.98
		2	1	3	0	0	4	\$ 145,522.98
		3	1	3	0	0	4	\$ 145,522.98
		3	1	4	0	0	4	\$ 145,522.98
		2	2	3	0	0	4	\$ 145,522.98
		2	2	4	0	0	4	\$ 145,522.98
		3	2	3	0	0	4	\$ 145,522.98
		3	2	4	0	0	4	\$ 145,522.98
		2	3	3	0	0	4	\$ 145,522.98
		3	3	3	0	0	4	\$ 145,522.98
		3	3	4	0	0	4	\$ 145,522.98
		3	4	0	0	0	4	\$ 145,522.98
		3	4	2	0	0	4	\$ 145,522.98
		2	4	3	0	0	4	\$ 145,522.98
		3	4	3	0	0	4	\$ 145,522.98
		3	4	4	0	0	4	\$ 145,522.98
		3	4	4	4	4	5	\$ 145,522.98

For each building type and building resistance, the reconstruction costs were estimated for each of the 197 damage states enumerated in Table C.1. The damage states were then averaged in the groups shown in Table C.1 to obtain the cost to reconstruct each of the 31 final damage states. 82 of the final 197 damage states caused “total” damage to the building, and were grouped into damage state 31.



## APPENDIX D

### WIND LOAD CALCULATIONS

The loss model followed the method for calculating wind loads used by the Florida Public Hurricane Loss Model (FPHLM) found in Engineering Volume II, Chapter 6. The typical home model used by the FPHLM shown in Figure 6-4 of the FPHLM Engineering Volume II. The home model used in this research, a description can be found in Appendix A, is very similar to the FPHLM model. Using the FPHLM method for estimating damage, the wind load and damage calculations happen in four steps. First, an initial load is calculated on all of the structural elements based on the wind speed. Second, and the structural elements are checked for failure under the initial loads. If the building envelope is compromised due to the failure of a window, wall, or portion of the roof, then the load on the structural elements is recalculated to account for the change in wind pressure on and inside the building. FPHLM uses calculations defined by ASCE-7 98 to calculate wind loads for each of the elements.

#### ***Roof Sheathing***

##### Initial Load

The wind pressure on the roof is calculated using the calculations from ASCE 7-98. First, the wind pressure in the three roof pressure zones is calculated. The appropriate Gust Factor and External Pressure Coefficient ( $GC_p$ ) is identified for each zone. The correct  $GC_p$  values are taken from the ASCE 7-98 Components and Cladding Pressure

Coefficient values and can be found in Table D.1, and the pressure is calculated using the wind speed,  $w$ , and initial internal pressure,  $GC_{pi}$  of 0.18 and the equations:

$$q_h = 0.00256 * 0.85 * w^2 \text{ (lb/ft}^2\text{)} \quad (D.1)$$

$$p = q_h[(GC_p) - (GC_{pi})] \text{ (lb/ft}^2\text{)} \quad (D.2)$$

The resulting wind pressure on each roof sheathing panel is then calculated by identifying how much of each panel is located in each zone and taking the aggregate pressure.

Table D.1 Roof zone gust factor and external pressure coefficients

<b>Roof Zone Components &amp; Cladding Pressure Coefficient Values</b>	
<b>Zone</b>	<b><math>GC_p</math></b>
1	-0.9
2	-1.7
3	-2.6

#### Initial Failure Check

The roof sheathing panels are checked for initial failure by comparing the load on each panel to the strength of each panel.

#### Final Load

If the building envelope is compromised and the internal pressure changes, then the load on the sheathing panels is recalculated using the same process outlined in the Initial Load calculations, substituting the use of Equation D.3 for Equation D.2.

$$p = q_h(0.8)GC_p - p_{in} \left( \frac{lb}{ft^2} \right) \quad (D.3)$$

### Final Failure Check

The roof sheathing panels are checked for final failure by comparing the load on each panel to the strength of each panel.

### ***Roof-to-Wall Connections***

#### Initial Load

The loads on the roof-to-wall connections are directly related to the loads on the roof sheathing. Roof trusses are spaced at 2 ft on center. Each roof-to-wall connection experiences the load of the roof sheathing panels in the tributary area of the corresponding roof truss. The tributary area is assumed to be on that side of the roof, one foot on either side of the roof truss.

#### Initial Failure Check

The roof-to-wall connections are checked for initial failure by comparing the load on each connection to the strength of each connection.

#### Final Load

If the building envelope is compromised and the internal pressure changes, then the load on the roof-to-wall connections is recalculated using the same process outlined in the Initial Load calculations.

### Final Failure Check

The roof-to-wall connections are checked for final failure by comparing the load on each connection to the strength of each connection. If a roof-to-wall connection fails, the load on that connection must be transferred to the adjacent connections, and those connections must be rechecked for failure. This process iterates until one full wall of roof to wall connections has failed (considered complete roof failure) or until the intact roof-to-wall connections can handle the applied load.

### ***Openings***

#### Initial Load

The wind pressure on the openings is calculated using the calculations from ASCE 7-98. First, the appropriate Gust Factor and External Pressure Coefficient ( $GC_p$ ) is identified for each wall. The correct  $GC_p$  values are taken from the ASCE 7-98 Components and Cladding Pressure Coefficient values and can be found in Table D.2. Figure D.1 displays the walls according to the direction of the wind. The pressure is calculated using the wind speed,  $w$ , and initial internal pressure,  $GC_{pi}$  of 0.18 and the Equations D.1 and D.2.

The resulting wind pressure on each opening is then calculated by identifying on which wall each opening is located. For front and back doors, when the doors are located on a side wall (perpendicular to the wind direction), assume that the front and back doors are not near the edge pressure areas where the pressure changes. For garage doors, when the door is located on a side wall, assume that the garage door is

$1/8^{\text{th}}$  in the edge pressure area where the pressure changes,  $7/8^{\text{th}}$  in the regular side pressure area. For windows designated as in the edge section are assumed to be  $1/4$  in the edge pressure area where the pressure changes, and  $3/4$  in the regular side pressure area.

Table D.2 Wall gust factor and external pressure coefficients

<b>Wall Components &amp; Cladding Pressure Coefficient Values</b>	
<b>Wall</b>	<b><math>GC_p</math></b>
Windward	1
Side Leading Edge	-1.4
Side	-1.1
Leeward	-0.8

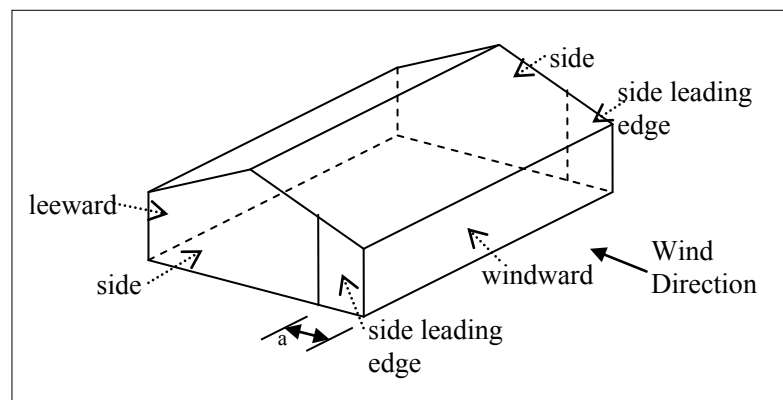


Figure D.1 Walls based on the wind direction corresponding to the gust effect factor and external pressure coefficients

### Initial Failure Check

The openings are checked for initial failure by comparing the load on each opening to the strength of that opening. Additionally, windborne debris is simulated and windows are checked for impact failures. The windborne debris is simulated using a process

defined by Chapter 4 in the FPHLM Engineering Volume II. The procedure is outlined briefly below but is described in detail in FPHLM Engineering Volume II.

The probability of damage to an opening  $p_D(V)$  given wind angle, wind speed, window size and wall size is defined in Equation D.4.

$$p_D(V) = 1 - \exp [-A * N_A * B * C * D] \quad (D.4)$$

Where  $A$  is the fraction of missile objects in the air, represented as a normal distribution function with mean at 135 mph and a standard deviation of 15 mph;  $N_A$  is the total number of available missile objects (default value of 100);  $B$  is the fraction of airborne missile objects that hit the house, represented as a linear function  $B = 0.002 * W - 0.10$ ;  $C$  is the fraction of the impact wall that is glass;  $D$  probability that the impacting missiles have enough momentum to damage the window, represented as a normal cumulative distribution function with mean 70 mph and standard deviation of 10 mph.

For each window on a windward wall,  $x_o$  is a sample from a uniform distribution  $x_o \sim U(1,100)$  representing the likelihood of windborne debris impact on that window. Compare the sampled value to the calculated value for  $p_D(V)$ , any window with a sampled value less than  $p_D(V)$  is assumed to fail under impact load.

### Final Load

If the building envelope is compromised and the internal pressure changes, then the load on the openings is recalculated using the same process outlined in the Initial Load calculations, substituting the use of Equation D.3 for Equation D.2.

### Final Failure Check

The openings are checked for final failure by comparing the load on each opening to the strength of each connection.

## ***Walls***

### Initial Load

Walls are subject to three types of loads: shear, uplift, and out of plane. The shear load from the wind pressure is calculated using the equations from ASCE 7-98. First, the appropriate External Pressure Coefficient ( $C_p$ ) is identified for each wall. The correct  $C_p$  values are taken from the ASCE 7-98 Main Wind Force Resisting System External Pressure Coefficient values and can be found in Table D.3. Figure D.2 identifies the location of the building surfaces in Table D.3. For wind orientations (Figure 6) perpendicular to the roof ridge ( $\phi = 0, 180$  degrees), use the coefficients from Figure D.2 (A). For wind orientations parallel to the roof ridge ( $\phi = 90, 270$  degrees), use the coefficients from Figure D.2 (B). For wind orientations at an angle to the ridge ( $\phi = 45, 135, 215, 325$  degrees), use the coefficients from Figure D.2 (A) and Figure D.2 (B) assuming the two “windward” walls are the walls meeting at the windward corner, and assuming the two “leeward” walls are the opposite two walls.

The pressure is calculated using the wind speed,  $w$ , a gust effect factor,  $G$ , of 0.85, and initial internal pressure,  $GC_{pi}$  of 0.18 and the Equations D.1 and D.2. If the wind orientation is at an angle to the ridge ( $\theta = 45, 135, 215, 325$ ) then reduce  $p$  by a factor 0.85 ( $K_d$ ) to account for the direction of the wind.

Table D.3 Wall main wind force resistance system external pressure coefficients

MWFRS Pressure Coefficient Values, $C_p$	
building surface	angle
	5 on 12
1	0.800
2	-0.300
3	-0.700
4	-0.250
5	-0.570
6	-0.900
7	-0.500
8	-0.300

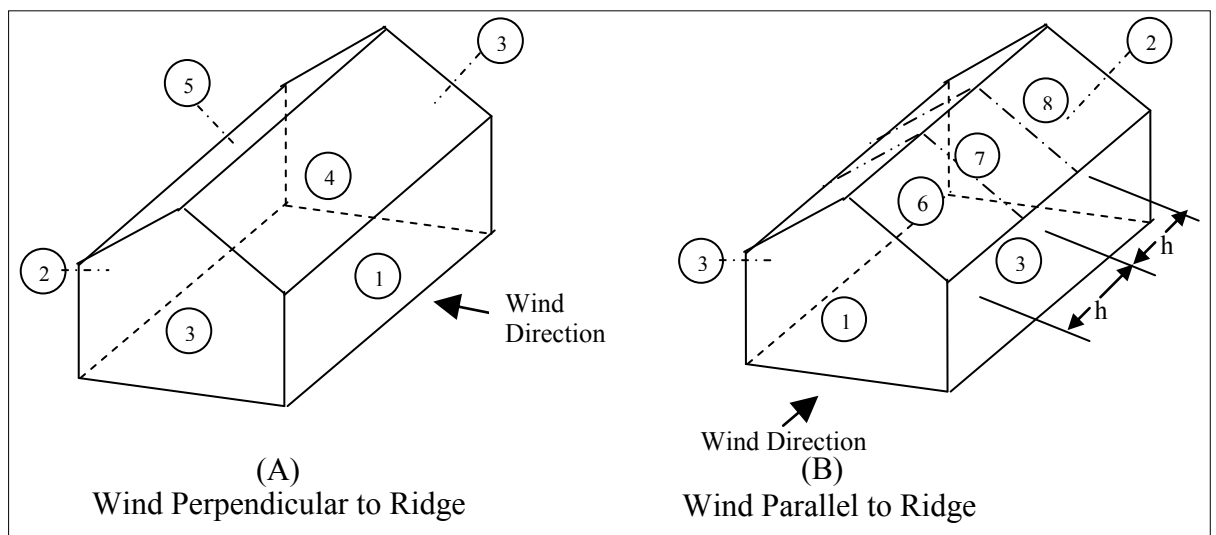


Figure D.2 Building surfaces corresponding to external pressure coefficients based on the wind direction



The resulting wind pressure on the walls is used to calculate the shear loads using the method outlined by ASCE 7-98. Using the following Equation D.5 and the diagrams in Figure D.3, the distributed wind pressure load,  $p$ , on the upper half of the wall is transferred to the roof beam. Half of the roof beam load is then transmitted to each of the two parallel shear walls as point loads  $V_1$  and  $V_2$ .

$$V = \frac{(p \frac{e}{2} l)}{2} \quad (lb) \quad (D.5)$$

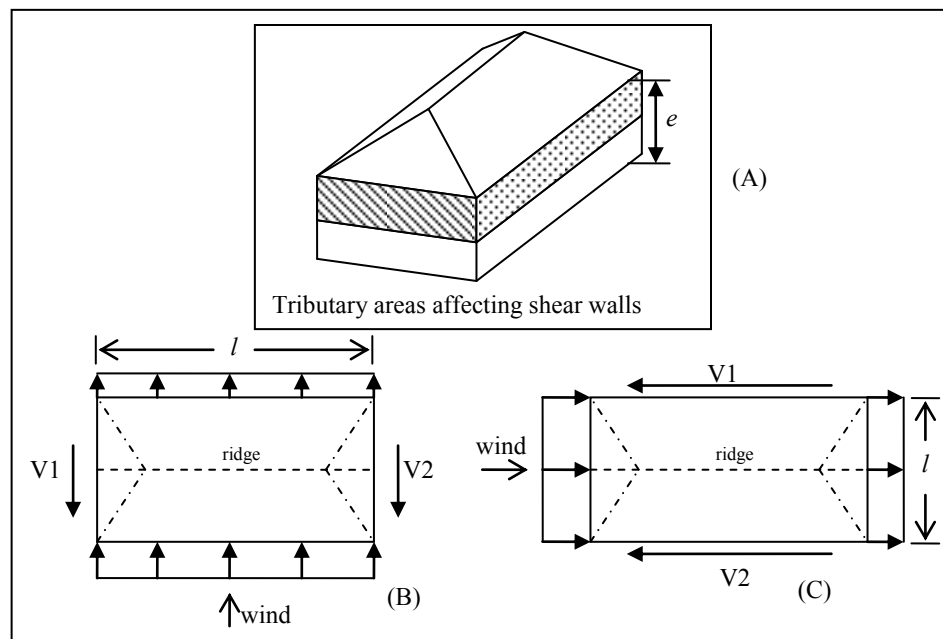


Figure D.3 Diagrams corresponding to the calculation of shear wall loads

The uplift load on the walls are associated with the uplift load from the roof-to-wall connections. The average uplift load on a wall is calculated using the mean of the roof-to-wall connection loads on that wall.

The out-of-plane loads for the walls use the calculations from ASCE 7-98. First, the appropriate Gust Factor and External Pressure Coefficient ( $GC_p$ ) is identified for each

wall. The correct  $GC_p$  values are taken from the ASCE 7-98 Components and Cladding Pressure Coefficient values and can be found in Table D.2. Figure D.1 displays the walls according to the direction of the wind. The pressure is calculated using the wind speed,  $w$ , and initial internal pressure,  $GC_{pi}$  of 0.18 and the Equations D.1 and D.2.

The load on the lateral connections is calculated based on a tributary area given in the Figure 4-8 in the FPHLM Engineering Volume II, Chapter 4. The figure assumes one interior wall at the center of the wall in question. Equation D.6 is used to calculate the total load based on the tributary area.

$$TotalLoad = p * \left[ (L - h) * \left( \frac{h}{2} \right) \right] \quad (lb) \quad (D.6)$$

The Total Load is distributed evenly to the lateral connections at the base of each wall. The model building is assumed to have 12 wall-floor connections on the front and back walls, and 6 on each of the side walls.

#### Initial Failure Check

The walls are checked for initial failure in all three load types (shear, uplift, out-of-plane) by comparing each load type to the corresponding wall strength.

#### Final Load

If the building envelope is compromised and the internal pressure changes, then the load on the walls is recalculated using the same process outlined in the Initial Load calculations, substituting the use of Equation D.3 for Equation D.2.

### Final Failure Check

The walls are checked for final failure in all three load types (shear, uplift, out-of-plane) by comparing each load type to the corresponding wall strength.

### ***Roof Cover***

#### Initial Load

Roof cover failure is highly dependent on roof sheathing failure. However, no other structural element is dependent on roof cover for calculation of load or failure. For that reason, the load on the roof cover is only calculated during the final load calculation step.

#### Initial Failure Check

An initial load is not calculated for roof cover; therefore an initial failure check of roof cover cannot be performed.

#### Final Load

The uplift wind load on the roof cover is calculated using the same calculations as the roof sheathing. First, the wind pressure in the three roof pressure zones is calculated. The appropriate Gust Factor and External Pressure Coefficient ( $GC_p$ ) is identified for each zone. The correct  $GC_p$  values are taken from the ASCE 7-98 Components and Cladding Pressure Coefficient values and can be found in Table D.1, and the pressure is calculated using the wind speed,  $w$ , and initial internal pressure,  $GC_{pi}$  of 0.18 and

the Equations D.1 and D.2. If the internal pressure has changed, then substitute the use of Equation D.3 for Equation D.2.

The resulting wind uplift on each roof sheathing panel is then calculated by identifying how much of each roof cover unit is located in each zone and taking the aggregate pressure.

#### Final Failure Check

Roof cover elements over sheathing panels that have been damaged are automatically counted as failed. For the remaining cover elements, each roof cover element is checked for final failure by comparing the load on each section of roof cover to the strength of each section.

#### ***Internal Pressure Recalculation***

After the initial failure check, if no openings have failed, the internal pressure remains unchanged. If one or more openings are damaged, a new internal pressure is calculated as a weighted average of pressures at locations of failed openings. Windows, and front and back doors are given equal weight and garage doors are weighted 4 times other openings. Equation D.7 is used to calculate the new internal pressure.

$$p_{in} = \frac{4(gar)p_g + \sum_{i=1}^n p_i}{n+4(gar)} \left( \frac{lb}{ft^2} \right) \quad (D.7)$$

Where  $p_g$  is the pressure on the failed garage door;  $p_i$  is the pressure on the individual failed windows or doors;  $n$  is the total number of failed windows and doors; and  $gar$  is 1 if the garage door has failed, 0 otherwise.

## APPENDIX E

### WIND DAMAGE MODEL CALIBRATION

As discussed in Chapter 5, the wind damage model was based primarily on the Florida Public Hurricane Loss Model's damage estimation model and on the damage states and some component strengths from HAZUS-MH. The fragility curves generated by HAZUS-MH were used to validate the results from the wind damage model. Direct comparisons of fragility curves of one building type are displayed in Figures E.1 through E.12. Corresponding descriptions of the component level damage states can be found in Table 3. The building level damage states are a combination of component damage states for each damage level (1 through 4).

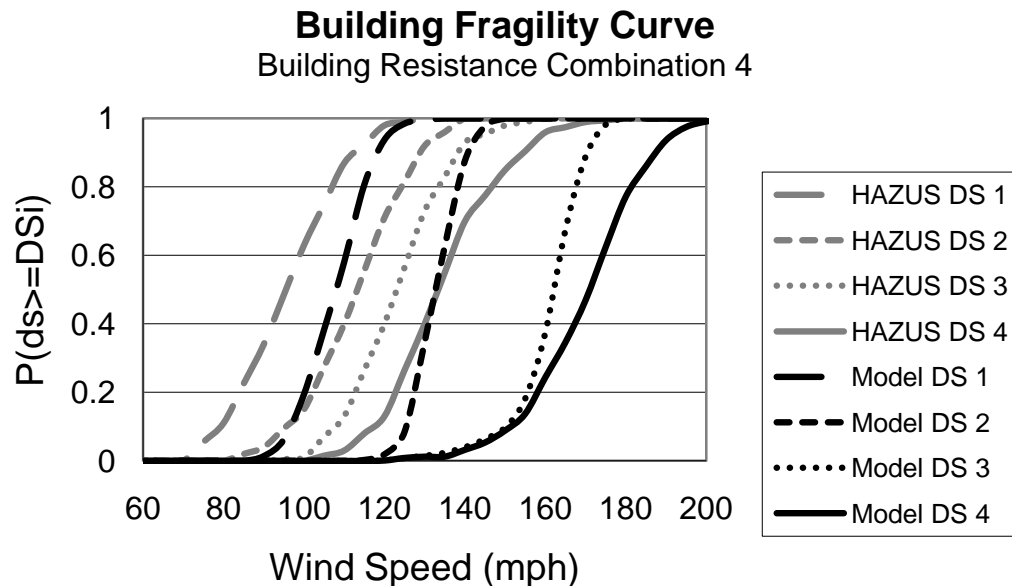


Figure E.1 Comparison of HAZUS and model fragility curves for building level damage states, for building resistance,  $c$ , 4

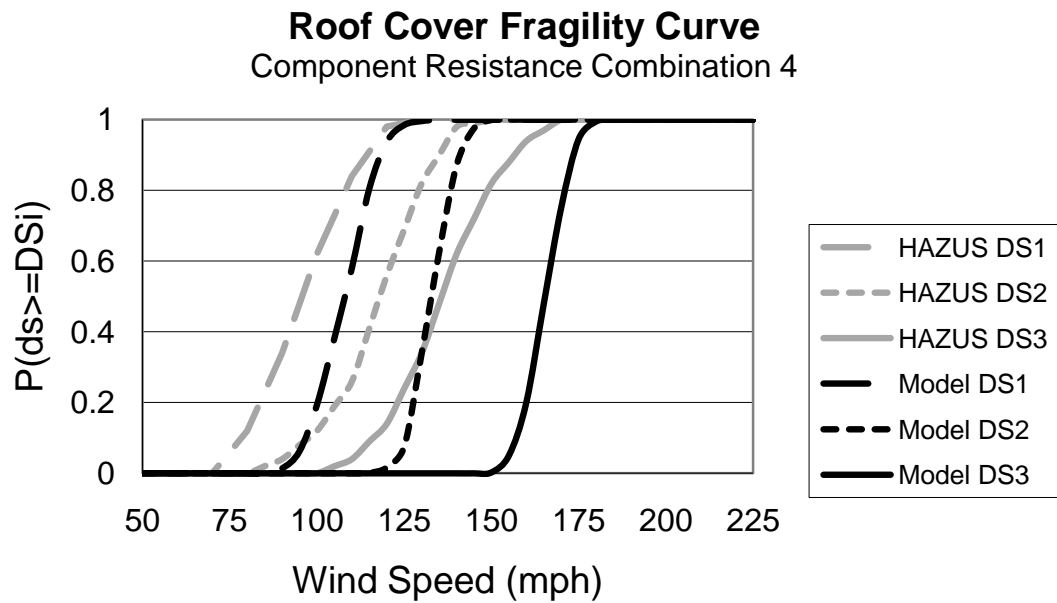


Figure E.2 Comparison of HAZUS and model fragility curves for roof cover damage states, for building resistance,  $c$ , 4

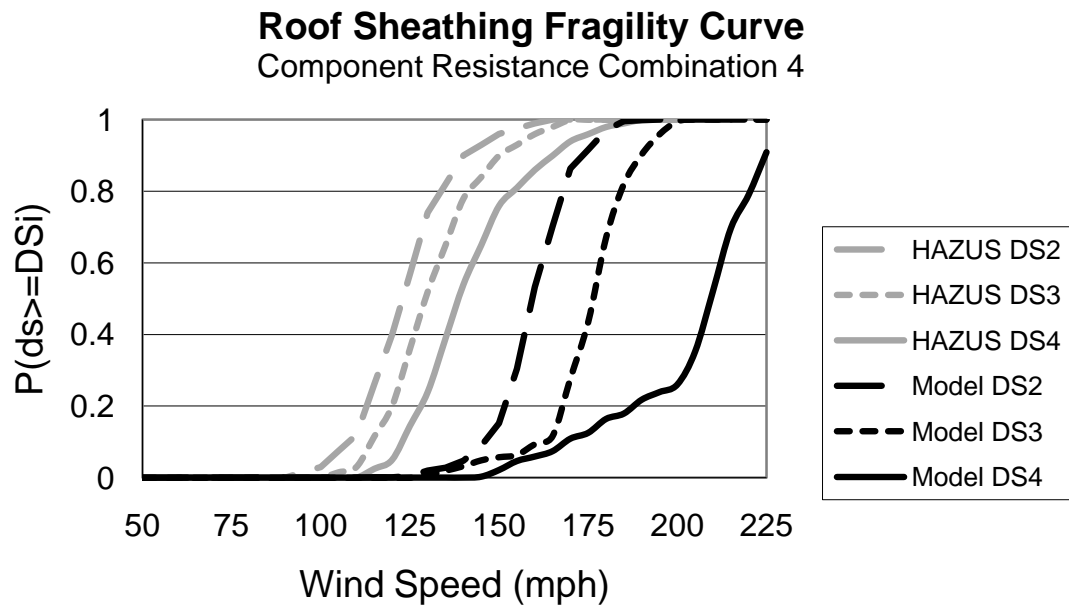


Figure E.3 Comparison of HAZUS and model fragility curves for roof sheathing damage states, for building resistance,  $c$ , 4

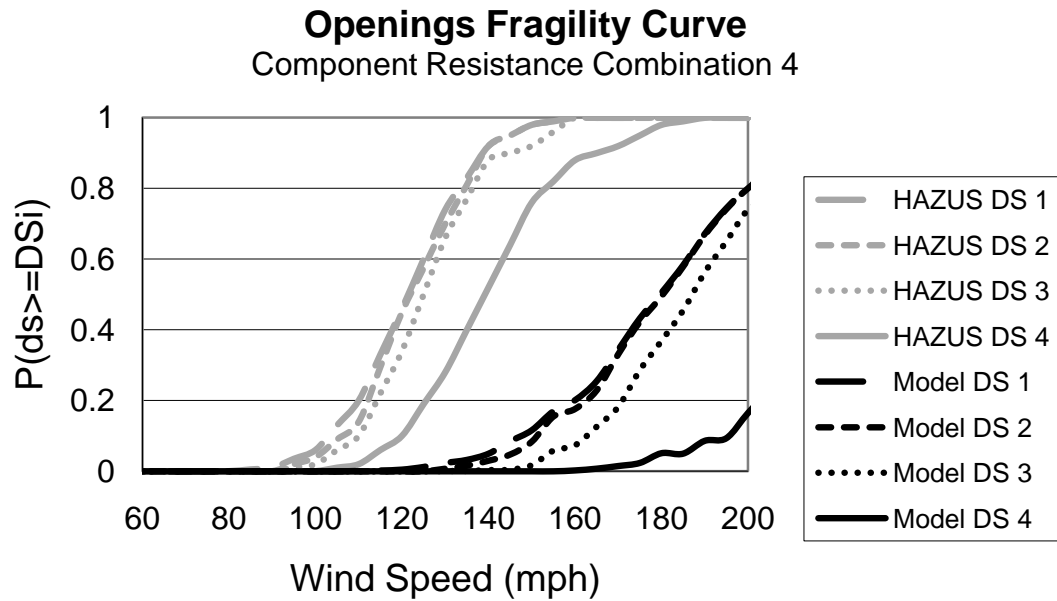


Figure E.4 Comparison of HAZUS and model fragility curves for opening damage states, for building resistance,  $c$ , 4

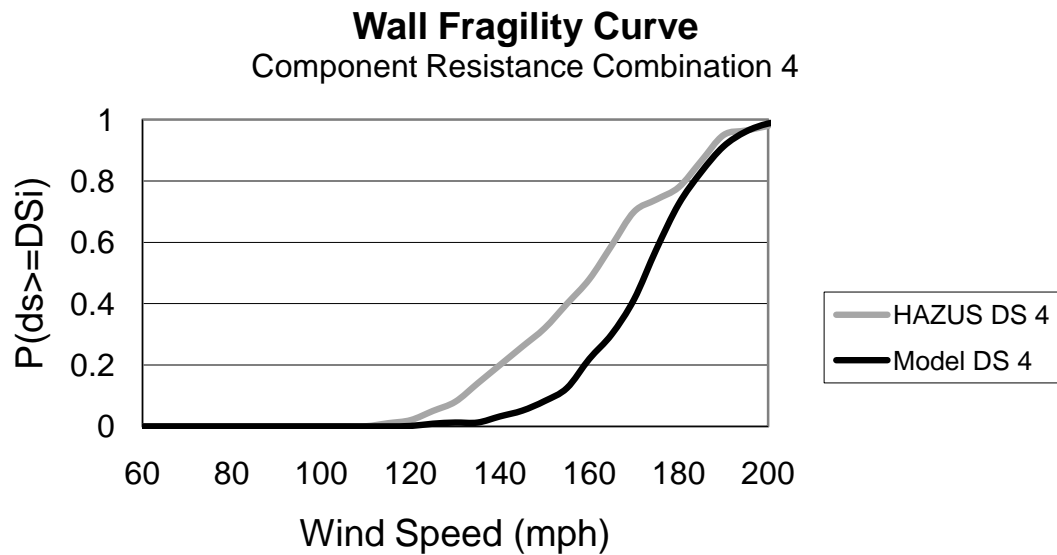


Figure E.5 Comparison of HAZUS and model fragility curves for wall damage states, for building resistance,  $c$ , 4



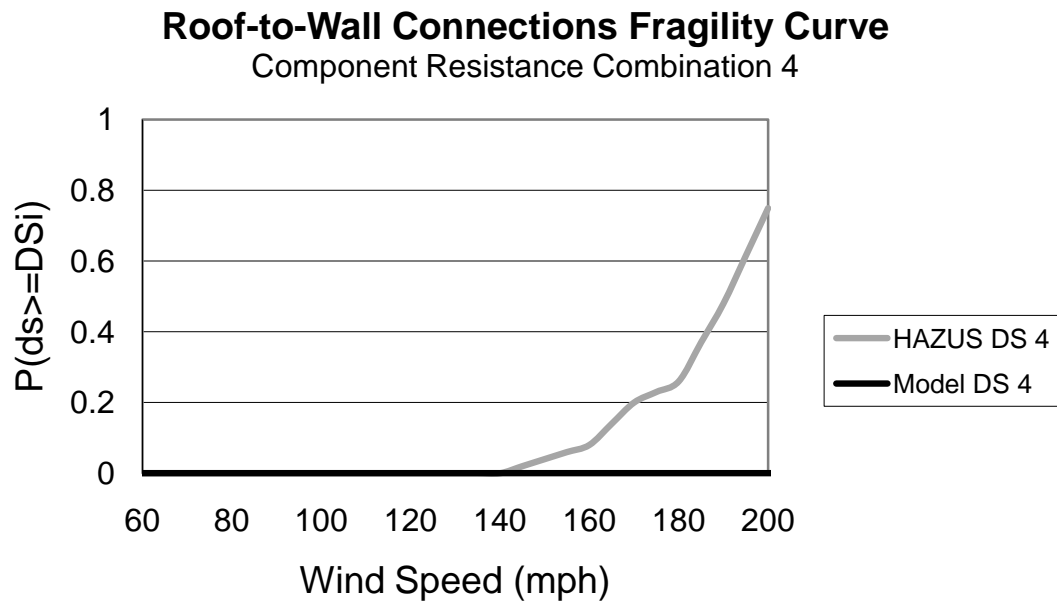


Figure E.6 Comparison of HAZUS and model fragility curves for roof-to-wall connection damage states, for building resistance,  $c$ , 4

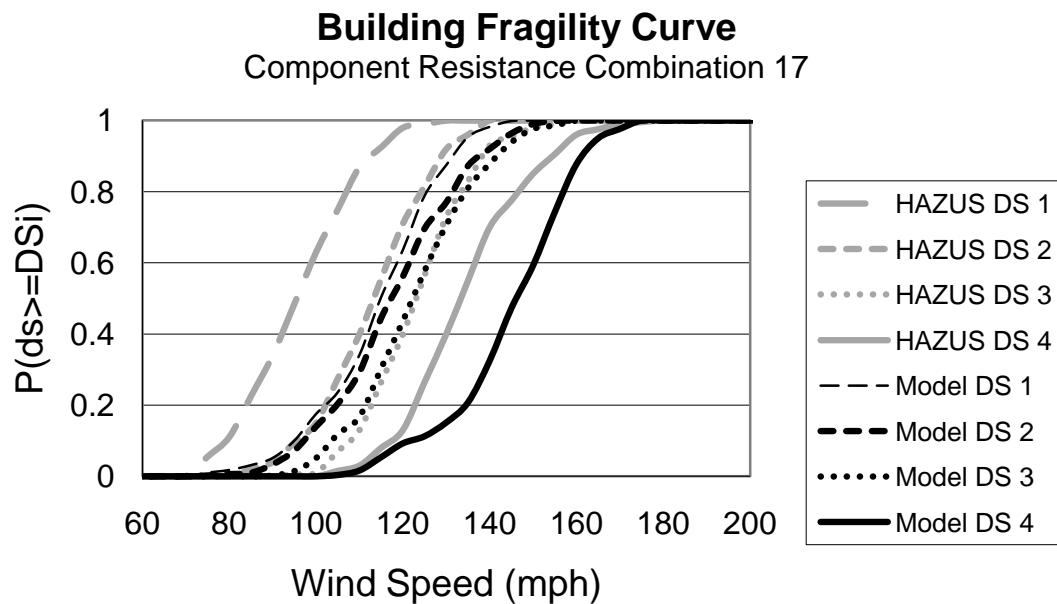


Figure E.7 Comparison of HAZUS and model fragility curves for building level damage states, for building resistance,  $c$ , 17

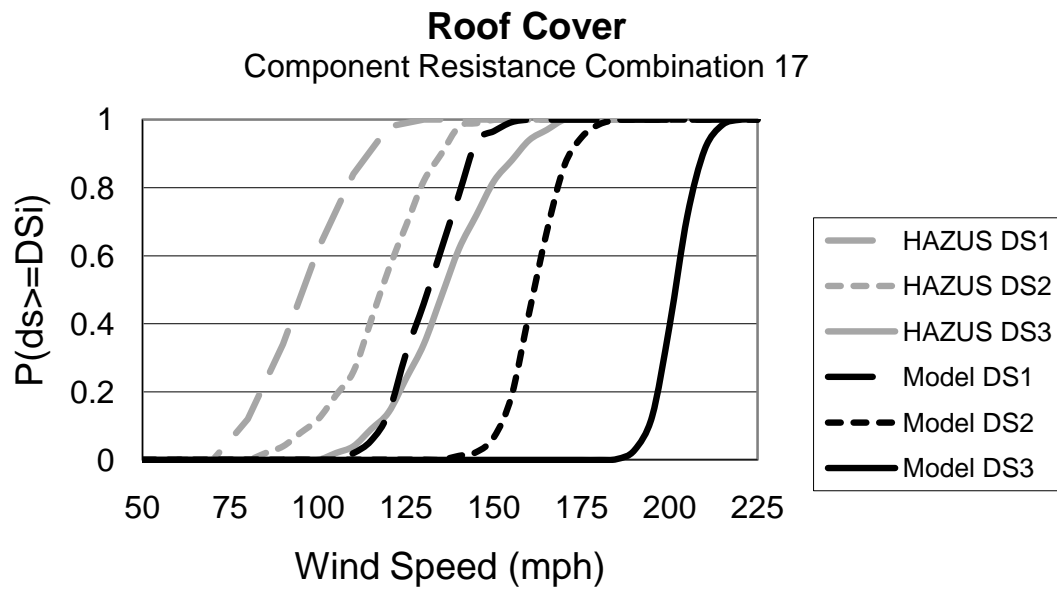


Figure E.8 Comparison of HAZUS and model fragility curves for roof cover damage states, for building resistance,  $c$ , 17

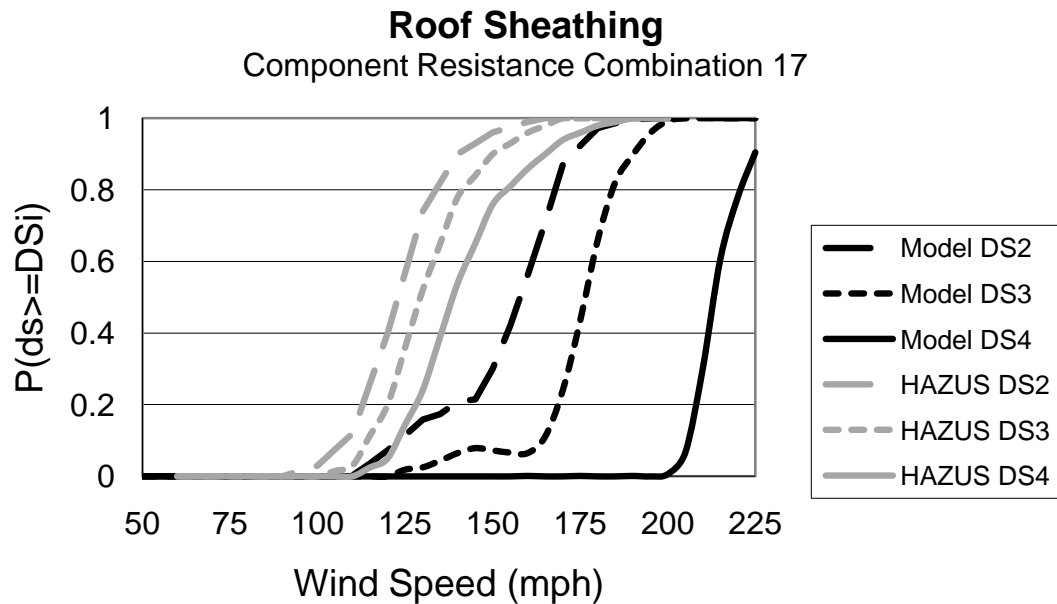


Figure E.9 Comparison of HAZUS and model fragility curves for roof sheathing damage states, for building resistance,  $c$ , 17

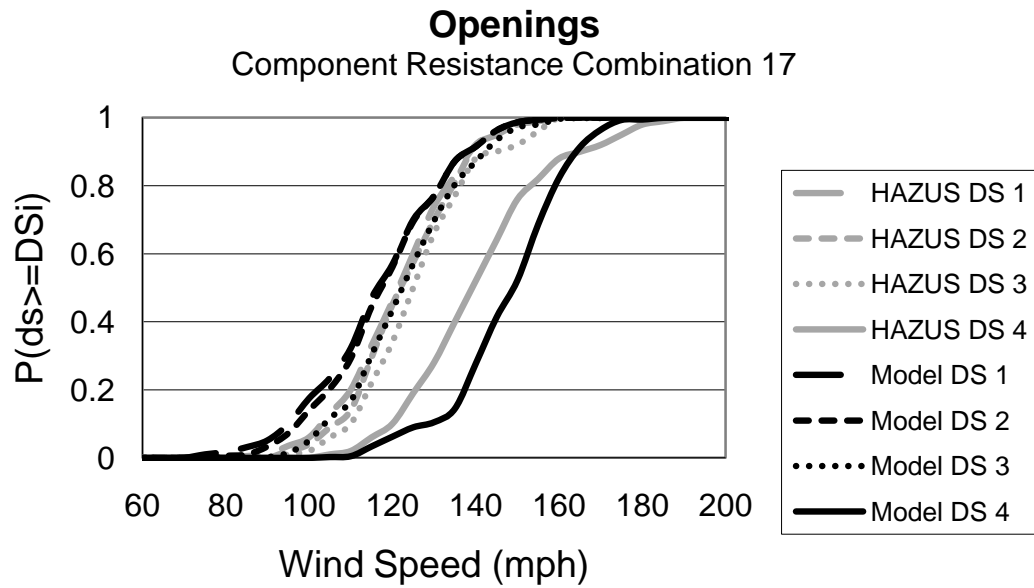


Figure E.10 Comparison of HAZUS and model fragility curves for opening damage states, for building resistance,  $c$ , 17

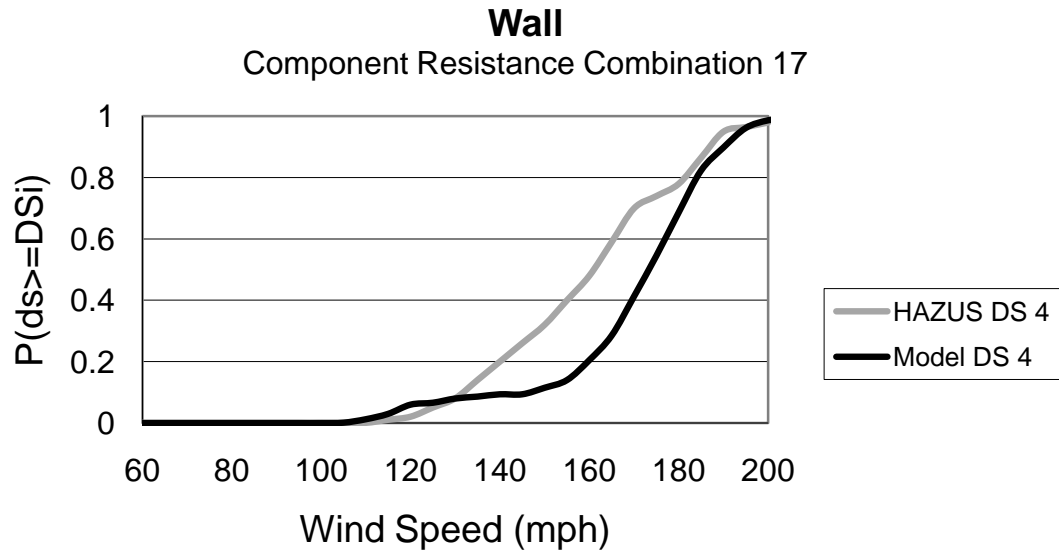


Figure E.11 Comparison of HAZUS and model fragility curves for wall damage states, for building resistance,  $c$ , 17

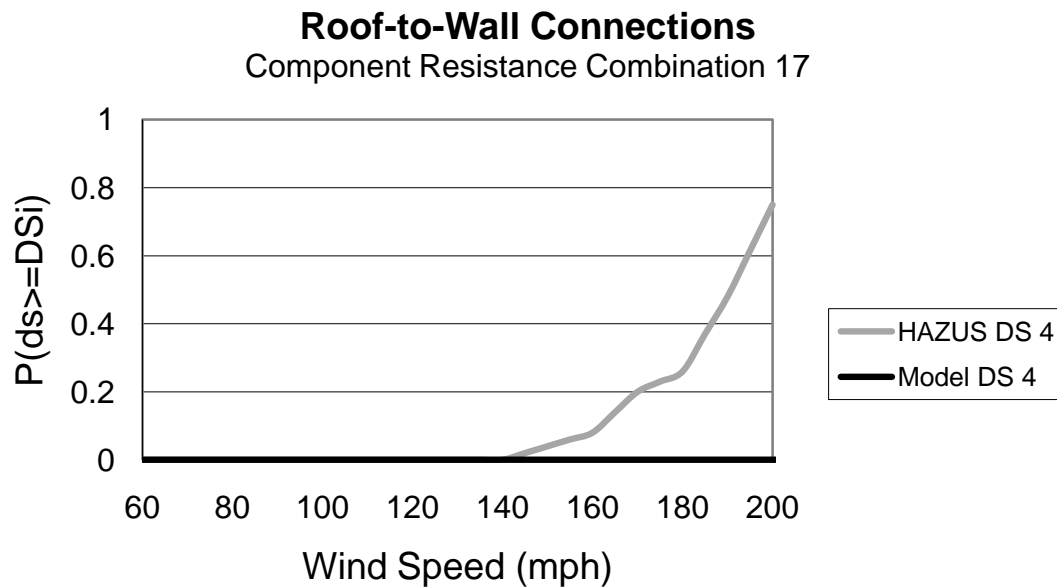


Figure E.12 Comparison of HAZUS and model fragility curves for roof-to-wall connection damage states, for building resistance,  $c$ , 17

The fragility curves were compared by estimating damage to similar buildings under similar conditions. The HAZUS fragility curves were taken from the HAZUS Technical Manual (FEMA 2003). In particular, the examples represented in Figures E.1 through E.12 are from Figure A.1 in FEMA 2003. The HAZUS fragility curves represent the damage to a one story building with 6d roof sheathing nails, strapped roof trusses, gable roof and no garage. The HAZUS curves are compared to two similar building types from the hazard model developed in this research. Both buildings are one story, gable roof with strapped trusses and no garage. The difference between building resistance 4 and building resistance 17 is the component strength configuration. Building resistance 4 has asphalt shingles, 8d roof sheathing nails, impact resistant openings, and gable end bracing. Building resistance 17 has high wind shingles, 8d roof sheathing nails, standard openings, and gable end bracing. The

resulting building fragility curves are similar in magnitude and slope for the wind damage model and HAZUS. The wind damage model curves for building resistance 4 are steeper, i.e. the number of buildings affected by a particular damage state escalates over a shorter change in wind speed. This difference can be primarily attributed to difference in the the storm definitions between the two models. The wind damage model developed for this research takes only one storm variable into account in the damage model, the maximum wind gust. HAZUS estimates damage based on several storm variables that define the character of the wind strength: central pressure difference, radius to maximum winds, heading, translation speed, and the coast crossing position or distance of closest approach. As a result, HAZUS results have a wider distribution of damage states from a given wind speed, and a more shallow fragility curve. Additional differences can be attributed to the differences in component configurations and strength.

## APPENDIX F

### FLOOD FRAGILITY CURVES

The flood fragility curves (probability of at least damage state  $d_6$  vs flood depth in m) were developed using the results of a flood loss simulation model for residential woodframe buildings described in Taggart (2007). Figures F.1 to F.8 present the fragility curves used in the model for one-story and two-story building categories, for each flood component configuration. For consistency with the rest of the resource allocation model, all flood depth values were translated into meters.

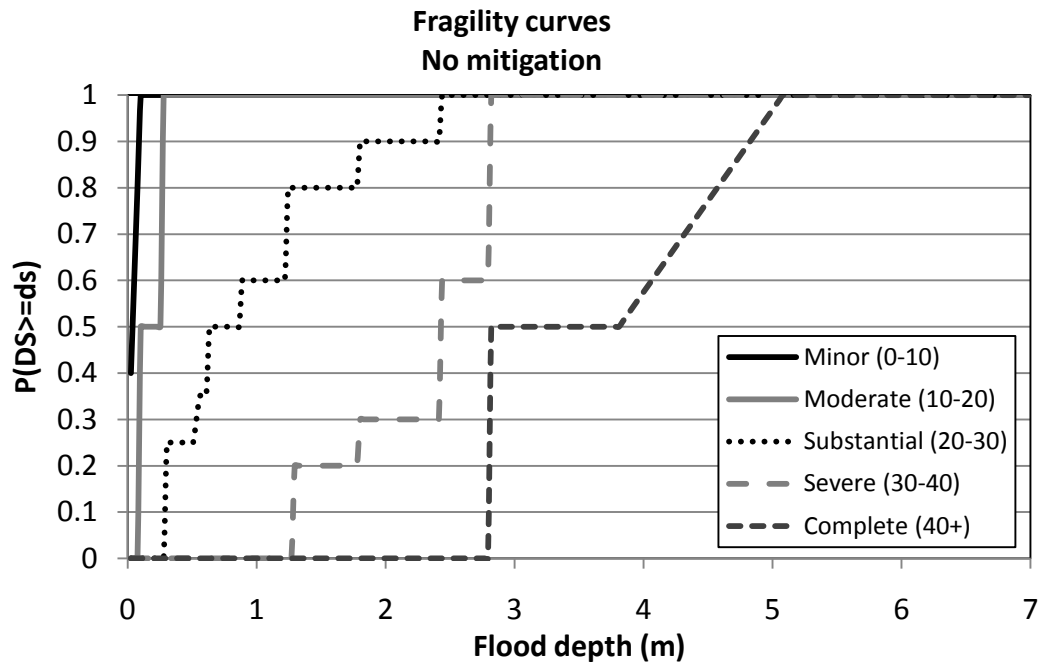


Figure F.1 Flood fragility curves for one-story building with no flood mitigation

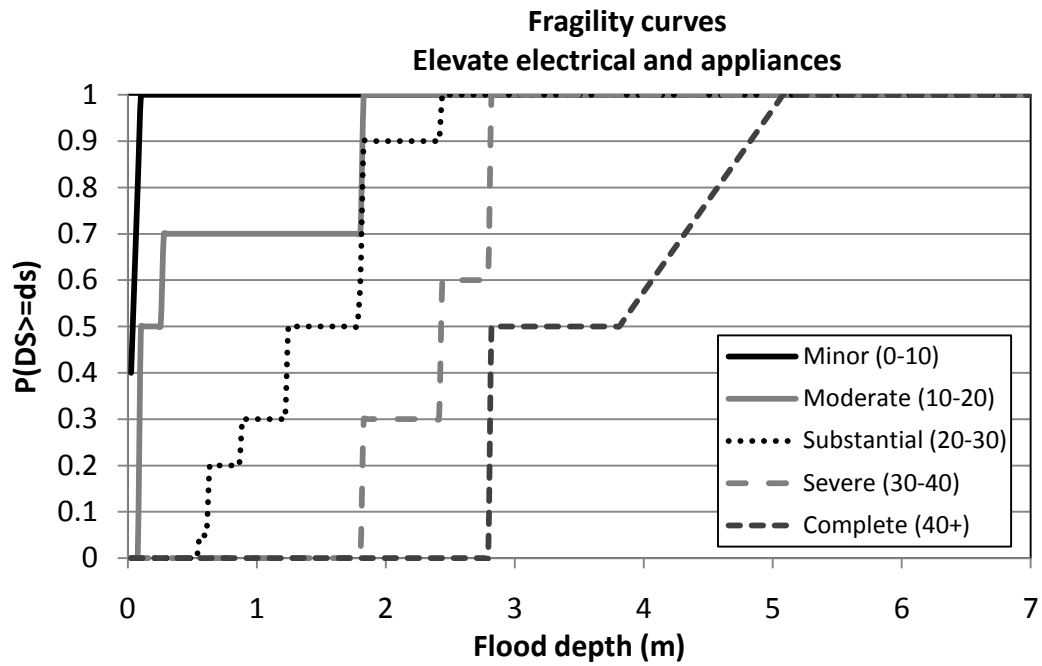


Figure F.2 Flood fragility curves for one-story building with elevated electrical and appliances

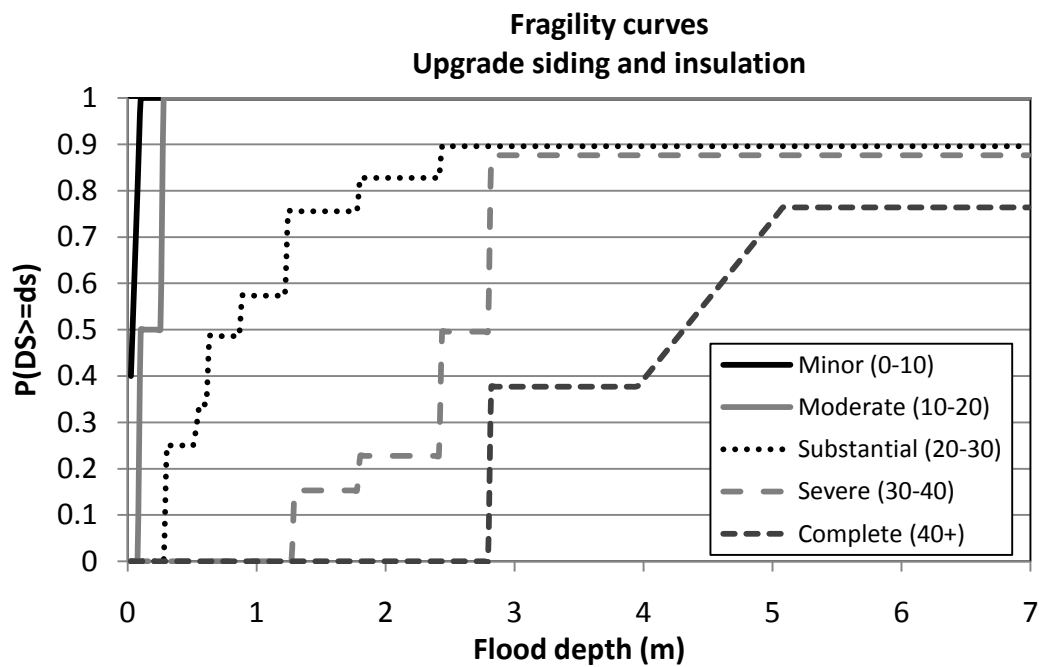


Figure F.3 Flood fragility curves for one-story building with upgraded siding and insulation

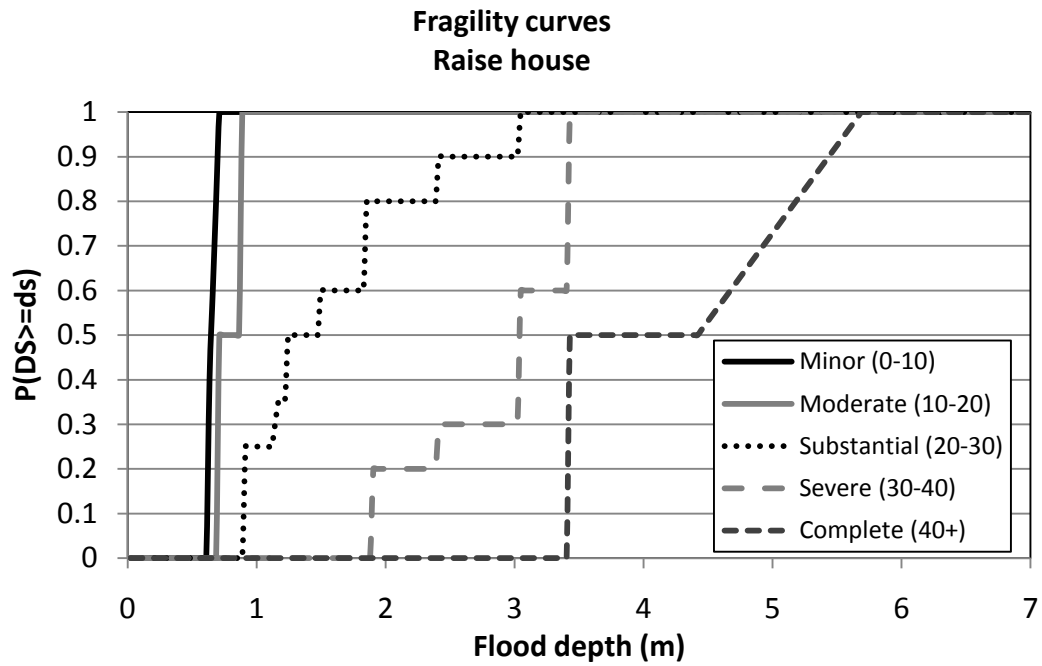


Figure F.4 Flood fragility curves for one-story raised building

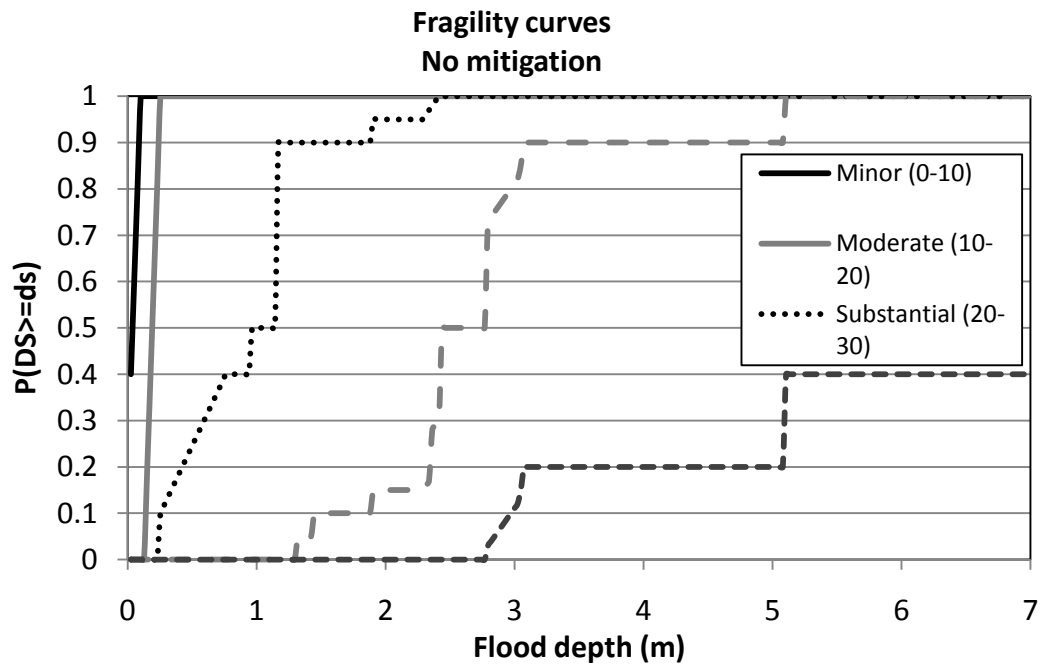


Figure F.5 Flood fragility curves for two-story building with no flood mitigation



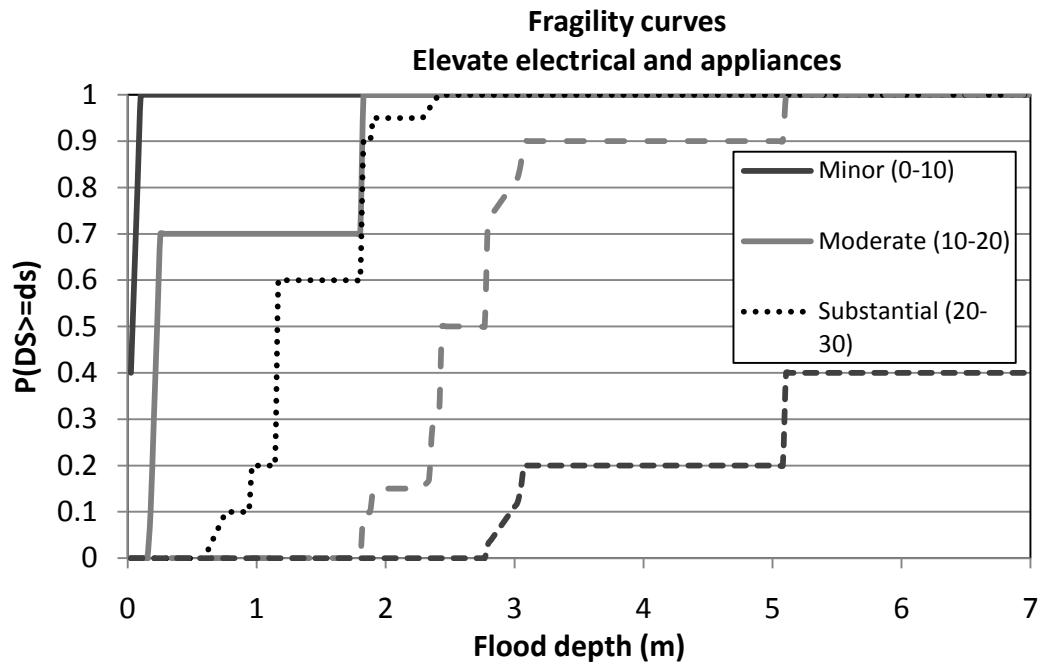


Figure F.6 Flood fragility curves for one-story building with elevated electrical and appliances

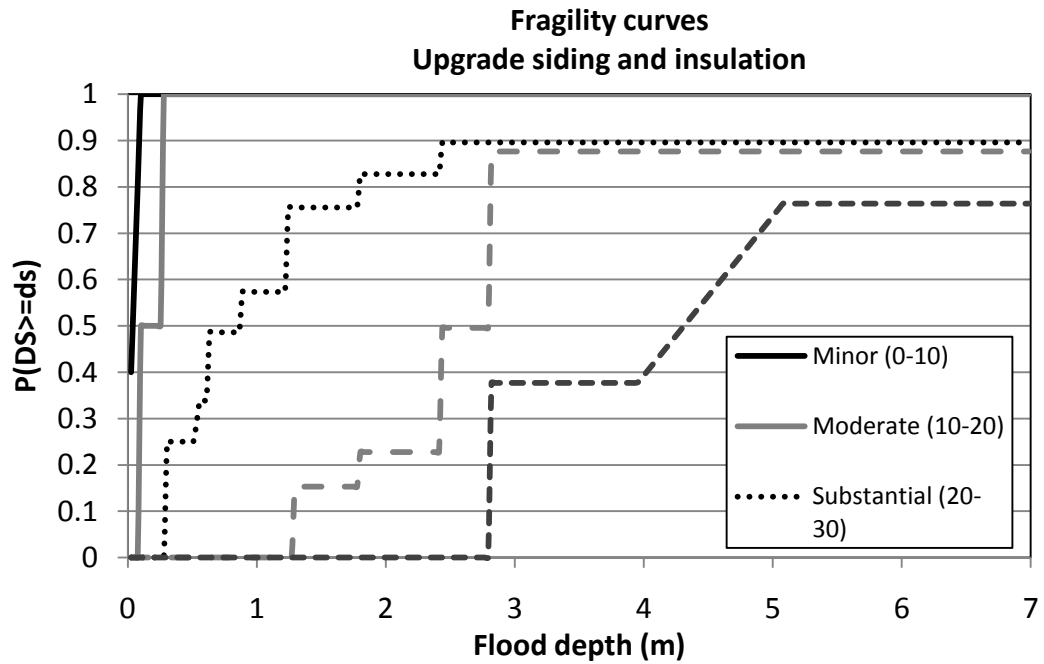


Figure F.7 Flood fragility curves for two-story building with upgraded siding and insulation

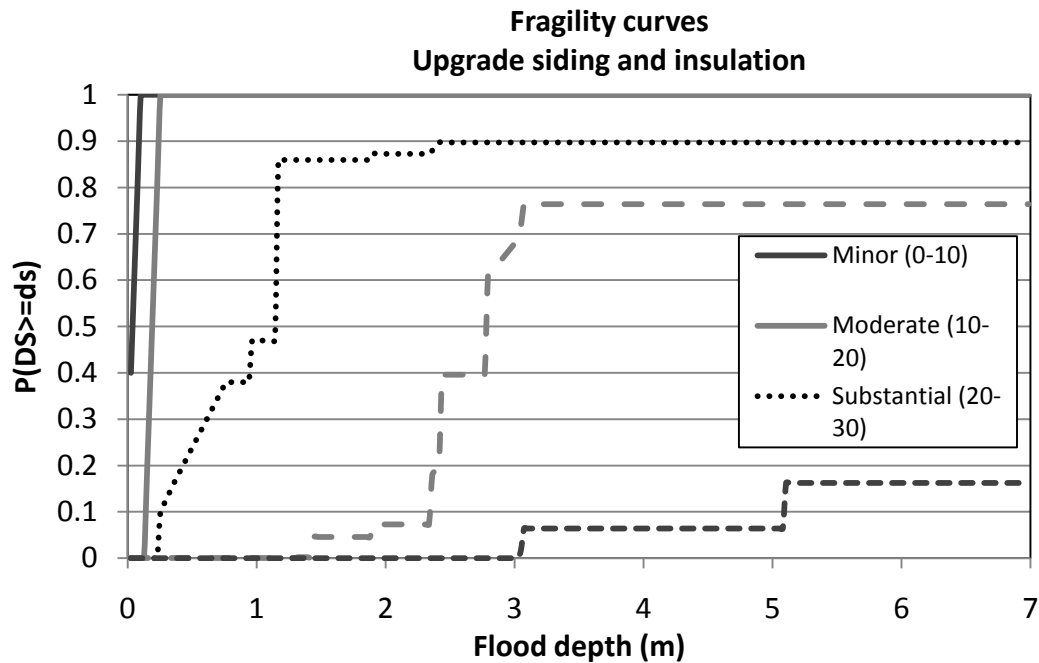


Figure F.8 Flood fragility curves for two-story raised building

Taggart (2007) provides median curves of flood depth (in) vs. percentage damage, for three building values (\$60, \$100, and \$160 per sq. ft.) and three flood durations (24, 96, and 168 hours). In Taggart (2007), Figure 31 provides those curves for one-story building with no basement and no flood mitigation, and Figure 64 does for two-story building with no basement and no flood mitigation. The one-story, no flood mitigation (flood component configuration 1 in Figure F.1) fragility curves were developed using the \$100/sq. ft. curves in Figure 31. Assuming that at a particular flood depth, the damage is uniformly distributed between the damage estimated for a 24-flood and the damage estimated for a 96-hour flood, the relative likelihoods of each flood damage state were estimated directly from the figure. The same was done for the two-story “no

flood mitigation” fragility curves using the average of the \$60/sq. ft. and \$100/sq. ft. curves and Figure 64 in Taggart (2007).

The fragility curves for the other flood component configurations were developed by modifying the “no flood mitigation” curves. The exact form of the modifications was based on the conclusions in Taggart (2007), which includes simulations for buildings with all four flood configurations. For flood component configuration 2 (elevated electrical and appliances), the fragility curves were assumed to be the same as the “no flood mitigation” curves at flood depths less than 4 in. and greater than 72 in.; and in between, all curves were shifted down by 0.3 while ensuring they remained monotonically increasing. For flood component configuration 3 (water-resistant foam wall insulation and vinyl siding), the fragility curves were assumed to be the same as the “no flood mitigation” curves at flood depths less than 14 in. Above 14 in., all curves were shifted down by 0.05 while ensuring they remain monotonically increasing. For flood component configuration 4 (entire house elevated 24 in.), the fragility curves were all shifted to the right by 24 in.

## APPENDIX G

### COST ESTIMATES

#### *Building Cost Estimate*

The cost to build the 8 model building types described in Appendix A was estimated using R.S. Means. To estimate the cost of the building using R.S. Means, the building was specified as an average class, detached single family wood frame home with 1800 square feet (3600 for two story) of living area. Additionally, the building was specified to have two bathrooms (1 full, 1 half), 11 windows (double for two story), and central air conditioning. Finally, the total cost of the building was multiplied by a location factor of 0.81 to represent construction costs in Wilmington, NC. The resulting cost estimates for the eight building types are shown in Table G.1.

Table G.1 Cost estimate for model building

<b>Building Type</b>	<b>Stories</b>	<b>Roof Type</b>	<b>Garage Type</b>	<b>Cost</b>
<b>1</b>	1	Hip	Garage	\$145,522
<b>2</b>	1	Gable	Garage	\$145,522
<b>3</b>	1	Hip	No Garage	\$148,253
<b>4</b>	1	Gable	No Garage	\$148,253
<b>5</b>	2	Hip	Garage	\$219,578
<b>6</b>	2	Gable	Garage	\$219,578
<b>7</b>	2	Hip	No Garage	\$222,309
<b>8</b>	2	Gable	No Garage	\$222,309

### ***Acquisition Costs***

The cost to acquire a building is assumed to be equal to the cost to construct the building plus 10% of the cost of the building. The additional 10% is added to the cost to reflect external costs to society, for example lost tax revenue. While this additional 10% factor could take on many values, 10% was chosen because it makes the choice to acquire a damaged building less attractive (more expensive) than rebuilding, unless the home is in an extremely high risk location. Table G.2 enumerates the input values used in the case study as the cost to acquire the eight building types.

Table G.2 Estimated Acquisition Costs

<b>Building Type</b>	<b>Stories</b>	<b>Roof Type</b>	<b>Garage Type</b>	<b>Acquisition Cost</b>
<b>1</b>	1	Hip	Garage	\$ 160,075
<b>2</b>	1	Gable	Garage	\$ 160,075
<b>3</b>	1	Hip	No Garage	\$ 163,079
<b>4</b>	1	Gable	No Garage	\$ 163,079
<b>5</b>	2	Hip	Garage	\$ 241,537
<b>6</b>	2	Gable	Garage	\$ 241,537
<b>7</b>	2	Hip	No Garage	\$ 244,540
<b>8</b>	2	Gable	No Garage	\$ 244,540

### ***Mitigation Costs***

The mitigation costs for each of the possible 124 mitigation alternatives were estimated using R.S. Means construction cost estimation. All cost estimates include labor, overhead and profit and are multiplied by a location factor of 0.81 for construction costs in Wilmington, NC. Tables G.3, G.4, G.5, G.6, G.7, and G.8 give the unit costs used for estimation of mitigation cost.

The cost estimate for a mitigation alternative was calculated by summing the cost to mitigate each of the alternatives individual components. For example, to mitigate from building resistance C1 [RC1, RS1, RWC1, O1, W1, F1] to C8 [RC2, RS2, RWC3, O3, W2, F1], the total cost is the combined cost to mitigate each of the roof cover, roof sheathing, roof-to-wall connections, openings, and wall components. Table G.8 displays enumerates the input values for the cost to perform all 124 mitigation alternatives for all eight building types.

Table G.3 Roof Cover Cost

Roof Cover		
Type	Description	Cover Cost (\$/ft <sup>2</sup> )
RC 1	Asphalt	\$1.07
RC 2	High Wind	\$1.85

Table G.4 Roof Sheathing Cost

Roof Sheathing		
Type	Description	Sheathing Cost (\$/ft <sup>2</sup> )
RS 1	Panels, 8d nails, 6/12"	\$1.06
RS 2	Panels, 8d nails, 3/6"	\$1.10
RS 3	Panels, 8d nails, 6/12", adhesive	\$1.48

Table G.5 Roof-to-Wall Connections Cost

Roof-to-Wall Connections		
Type	Description	RWC Cost (\$/ft <sup>2</sup> )
RWC 1	Toe-Nail Connection	\$1.54
RWC 2	Toe-Nail Connection & Adhesive	\$1.87
RWC 3	Toe-Nail Connection & Hurricane Strap	\$2.80

Table G.6 Openings Cost

<b>Openings</b>		
<b>Windows</b>	<b>Description</b>	<b>Opening Cost (\$/unit)</b>
O 1	Standard	\$425
O 2	Impact Resistant	\$602
O 3	Plywood Shutters	\$15
O 4	Hurricane Shutters	\$768
<b>Doors</b>	<b>Description</b>	<b>Opening Cost (\$/unit)</b>
O 1	Standard	\$135
O 2	Impact Resistant	\$313
O 3	Plywood Shutters	\$19
O 4	Hurricane Shutters	\$1,563

Table G.7 Wall Cost

<b>Walls</b>		
<b>Type</b>	<b>Description</b>	<b>Wall Cost (\$)</b>
W 1	No Gable End Bracing	\$0
W 2	Gable End Bracing	\$154

Table G.8 Flood Cost

<b>Flood Vulnerability</b>		
<b>1 Story</b>	<b>Description</b>	<b>Flood Cost (\$)</b>
F 1	None	\$0.00
F 2	Elevate Appliances and Electrical	\$2,000.00
F 3	Upgrade Insulation and Siding	\$17,901.00
F 4	Elevate Structure	\$126,000.00
<b>2 Story</b>	<b>Description</b>	<b>Flood Cost (\$)</b>
F 1	None	\$0.00
F 2	Elevate Appliances and Electrical	\$3,000.00
F 3	Upgrade Insulation and Siding	\$35,802.00
F 4	Elevate Structure	\$126,000.00

Table G.9 Estimated Mitigation Costs



Building Type	Initial Resistance	Final Resistance				
		8	20	27	45	61
1	1	\$6,234	\$2,000	\$8,234	\$24,135	\$145,523
2	1	\$6,343	\$2,000	\$8,343	\$24,244	\$145,523
3	1	\$6,234	\$2,000	\$8,234	\$24,135	\$148,253
4	1	\$6,343	\$2,000	\$8,343	\$24,244	\$148,253
5	1	\$6,234	\$3,000	\$9,234	\$42,036	\$219,579
6	1	\$6,343	\$3,000	\$9,343	\$42,145	\$219,579
7	1	\$6,234	\$3,000	\$9,234	\$42,036	\$222,309
8	1	\$6,343	\$3,000	\$9,343	\$42,145	\$222,309

Building Type	Initial Resistance	Final Resistance						
		3	21	22	39	40	55	56
1	2	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
2	2	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
3	2	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
4	2	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
5	2	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
6	2	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
7	2	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309
8	2	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309

Building Type	Initial Resistance	Final Resistance		
		22	40	56
1	3	\$2,000	\$17,901	\$145,523
2	3	\$2,000	\$17,901	\$145,523
3	3	\$2,000	\$17,901	\$148,253
4	3	\$2,000	\$17,901	\$148,253
5	3	\$3,000	\$35,802	\$219,579
6	3	\$3,000	\$35,802	\$219,579
7	3	\$3,000	\$35,802	\$222,309
8	3	\$3,000	\$35,802	\$222,309

Building Type	Initial Resistance	Final Resistance						
		5	23	24	41	42	57	58
1	4	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
2	4	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
3	4	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
4	4	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
5	4	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
6	4	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
7	4	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309
8	4	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309

Table G.9 (Continued)

Building Type	Initial Resistance	Final Resistance		
		24	42	58
1	5	\$2,000	\$17,901	\$145,523
2	5	\$2,000	\$17,901	\$145,523
3	5	\$2,000	\$17,901	\$148,253
4	5	\$2,000	\$17,901	\$148,253
5	5	\$3,000	\$35,802	\$219,579
6	5	\$3,000	\$35,802	\$219,579
7	5	\$3,000	\$35,802	\$222,309
8	5	\$3,000	\$35,802	\$222,309

Building Type	Initial Resistance	Final Resistance						
		8	25	27	43	45	59	61
1	6	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
2	6	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
3	6	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
4	6	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
5	6	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
6	6	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
7	6	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309
8	6	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309

Building Type	Initial Resistance	Final Resistance						
		9	26	28	44	46	60	62
1	7	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
2	7	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$145,523	\$145,523
3	7	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
4	7	\$3,613	\$2,000	\$5,613	\$17,901	\$21,514	\$148,253	\$148,253
5	7	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
6	7	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$219,579	\$219,579
7	7	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309
8	7	\$3,613	\$3,000	\$6,613	\$35,802	\$39,415	\$222,309	\$222,309

Building Type	Initial Resistance	Final Resistance						
		9	27	28	45	46	61	62
1	8	\$11,576	\$2,000	\$2,000	\$13,576	\$17,901	\$29,477	\$145,523
2	8	\$11,576	\$2,000	\$2,000	\$13,576	\$17,901	\$29,477	\$145,523
3	8	\$11,576	\$2,000	\$2,000	\$13,576	\$17,901	\$29,477	\$148,253
4	8	\$11,576	\$2,000	\$2,000	\$13,576	\$17,901	\$29,477	\$148,253
5	8	\$11,576	\$3,000	\$3,000	\$14,576	\$35,802	\$47,378	\$219,579
6	8	\$11,576	\$3,000	\$3,000	\$14,576	\$35,802	\$47,378	\$219,579
7	8	\$11,576	\$3,000	\$3,000	\$14,576	\$35,802	\$47,378	\$222,309
8	8	\$11,576	\$3,000	\$3,000	\$14,576	\$35,802	\$47,378	\$222,309

Table G.9 (Continued)

Building Type	Initial Resistance	Final Resistance		
		28	46	62
1	9	\$2,000	\$17,901	\$145,523
2	9	\$2,000	\$17,901	\$145,523
3	9	\$2,000	\$17,901	\$148,253
4	9	\$2,000	\$17,901	\$148,253
5	9	\$3,000	\$35,802	\$219,579
6	9	\$3,000	\$35,802	\$219,579
7	9	\$3,000	\$35,802	\$222,309
8	9	\$3,000	\$35,802	\$222,309

Building Type	Initial Resistance	Final Resistance		
		29	47	63
1	10	\$2,000	\$17,901	\$145,523
2	10	\$2,000	\$17,901	\$145,523
3	10	\$2,000	\$17,901	\$148,253
4	10	\$2,000	\$17,901	\$148,253
5	10	\$3,000	\$35,802	\$219,579
6	10	\$3,000	\$35,802	\$219,579
7	10	\$3,000	\$35,802	\$222,309
8	10	\$3,000	\$35,802	\$222,309

Building Type	Initial Resistance	Final Resistance						
		13	30	32	48	50	64	66
1	11	\$16,517	\$2,000	\$18,517	\$17,901	\$34,418	\$145,523	\$145,523
2	11	\$16,517	\$2,000	\$18,517	\$17,901	\$34,418	\$145,523	\$145,523
3	11	\$16,517	\$2,000	\$18,517	\$17,901	\$34,418	\$148,253	\$148,253
4	11	\$16,517	\$2,000	\$18,517	\$17,901	\$34,418	\$148,253	\$148,253
5	11	\$16,517	\$3,000	\$19,517	\$35,802	\$52,319	\$219,579	\$219,579
6	11	\$16,517	\$3,000	\$19,517	\$35,802	\$52,319	\$219,579	\$219,579
7	11	\$16,517	\$3,000	\$19,517	\$35,802	\$52,319	\$222,309	\$222,309
8	11	\$16,517	\$3,000	\$19,517	\$35,802	\$52,319	\$222,309	\$222,309

Building Type	Initial Resistance	Final Resistance		
		31	49	65
1	12	\$2,000	\$17,901	\$145,523
2	12	\$2,000	\$17,901	\$145,523
3	12	\$2,000	\$17,901	\$148,253
4	12	\$2,000	\$17,901	\$148,253
5	12	\$3,000	\$35,802	\$219,579
6	12	\$3,000	\$35,802	\$219,579
7	12	\$3,000	\$35,802	\$222,309
8	12	\$3,000	\$35,802	\$222,309

Table G.9 (Continued)

Building Type	Initial Resistance	Final Resistance		
		32	50	66
1	13	\$2,000	\$17,901	\$145,523
2	13	\$2,000	\$17,901	\$145,523
3	13	\$2,000	\$17,901	\$148,253
4	13	\$2,000	\$17,901	\$148,253
5	13	\$3,000	\$35,802	\$219,579
6	13	\$3,000	\$35,802	\$219,579
7	13	\$3,000	\$35,802	\$222,309
8	13	\$3,000	\$35,802	\$222,309

Building Type	Initial Resistance	Final Resistance								
		15	16	17	18	33	34	35	36	37
1	14	\$7,242	\$176	\$263	\$11,576	\$2,000	\$9,242	\$2,176	\$2,263	\$13,576
2	14	\$7,242	\$300	\$373	\$11,576	\$2,000	\$9,242	\$2,300	\$2,373	\$13,576
3	14	\$7,242	\$176	\$263	\$11,576	\$2,000	\$9,242	\$2,176	\$2,263	\$13,576
4	14	\$7,242	\$300	\$373	\$11,576	\$2,000	\$9,242	\$2,300	\$2,373	\$13,576
5	14	\$7,242	\$176	\$263	\$11,576	\$3,000	\$10,242	\$3,176	\$3,263	\$14,576
6	14	\$7,242	\$300	\$373	\$11,576	\$3,000	\$10,242	\$3,300	\$3,373	\$14,576
7	14	\$7,242	\$176	\$263	\$11,576	\$3,000	\$10,242	\$3,176	\$3,263	\$14,576
8	14	\$7,242	\$300	\$373	\$11,576	\$3,000	\$10,242	\$3,300	\$3,373	\$14,576

Building Type	Initial Resistance	Final Resistance						
		38	51	52	53	54	67	68
1	14	\$13,397	\$25,143	\$18,077	\$18,164	\$29,477	\$145,523	\$145,523
2	14	\$13,552	\$25,143	\$18,201	\$18,274	\$29,477	\$145,523	\$145,523
3	14	\$13,397	\$25,143	\$18,077	\$18,164	\$29,477	\$148,253	\$148,253
4	14	\$13,552	\$25,143	\$18,201	\$18,274	\$29,477	\$148,253	\$148,253
5	14	\$14,397	\$43,044	\$35,978	\$36,065	\$47,378	\$219,579	\$219,579
6	14	\$14,552	\$43,044	\$36,102	\$36,175	\$47,378	\$219,579	\$219,579
7	14	\$14,397	\$43,044	\$35,978	\$36,065	\$47,378	\$222,309	\$222,309
8	14	\$14,552	\$43,044	\$36,102	\$36,175	\$47,378	\$222,309	\$222,309

Building Type	Initial Resistance	Final Resistance		
		34	51	67
1	15	\$2,000	\$17,901	\$145,523
2	15	\$2,000	\$17,901	\$145,523
3	15	\$2,000	\$17,901	\$148,253
4	15	\$2,000	\$17,901	\$148,253
5	15	\$3,000	\$35,802	\$219,579
6	15	\$3,000	\$35,802	\$219,579
7	15	\$3,000	\$35,802	\$222,309
8	15	\$3,000	\$35,802	\$222,309

Table G.9 (Continued)

Building Type	Initial Resistance	Final Resistance	
		35	52
1	16	\$2,000	\$17,901
2	16	\$2,000	\$17,901
3	16	\$2,000	\$17,901
4	16	\$2,000	\$17,901
5	16	\$3,000	\$35,802
6	16	\$3,000	\$35,802
7	16	\$3,000	\$35,802
8	16	\$3,000	\$35,802

Building Type	Initial Resistance	Final Resistance	
		36	53
1	17	\$2,000	\$17,901
2	17	\$2,000	\$17,901
3	17	\$2,000	\$17,901
4	17	\$2,000	\$17,901
5	17	\$3,000	\$35,802
6	17	\$3,000	\$35,802
7	17	\$3,000	\$35,802
8	17	\$3,000	\$35,802

Building Type	Initial Resistance	Final Resistance		
		37	54	68
1	18	\$2,000	\$17,901	\$145,523
2	18	\$2,000	\$17,901	\$145,523
3	18	\$2,000	\$17,901	\$148,253
4	18	\$2,000	\$17,901	\$148,253
5	18	\$3,000	\$35,802	\$219,579
6	18	\$3,000	\$35,802	\$219,579
7	18	\$3,000	\$35,802	\$222,309
8	18	\$3,000	\$35,802	\$222,309

Building Type	Initial Resistance	Final Resistance							
		3	5	22	24	40	42	56	58
1	19	\$5,938	\$6,025	\$7,938	\$8,025	\$23,839	\$23,926	\$145,523	\$145,523
2	19	\$5,908	\$5,980	\$7,908	\$7,980	\$23,809	\$23,881	\$145,523	\$145,523
3	19	\$5,938	\$6,025	\$7,938	\$8,025	\$23,839	\$23,926	\$148,253	\$148,253
4	19	\$5,908	\$5,980	\$7,908	\$7,980	\$23,809	\$23,881	\$148,253	\$148,253
5	19	\$5,938	\$6,025	\$8,938	\$9,025	\$41,740	\$41,827	\$219,579	\$219,579
6	19	\$5,908	\$5,980	\$8,908	\$8,980	\$41,710	\$41,782	\$219,579	\$219,579
7	19	\$5,938	\$6,025	\$8,938	\$9,025	\$41,740	\$41,827	\$222,309	\$222,309
8	19	\$5,908	\$5,980	\$8,908	\$8,980	\$41,710	\$41,782	\$222,309	\$222,309

Table G.9 (Continued)

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		27			22
1	20	\$6,234	1	21	\$3,613
2	20	\$6,343	2	21	\$3,613
3	20	\$6,234	3	21	\$3,613
4	20	\$6,343	4	21	\$3,613
5	20	\$6,234	5	21	\$3,613
6	20	\$6,343	6	21	\$3,613
7	20	\$6,234	7	21	\$3,613
8	20	\$6,343	8	21	\$3,613

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		24			27
1	23	\$3,613	1	25	\$3,613
2	23	\$3,613	2	25	\$3,613
3	23	\$3,613	3	25	\$3,613
4	23	\$3,613	4	25	\$3,613
5	23	\$3,613	5	25	\$3,613
6	23	\$3,613	6	25	\$3,613
7	23	\$3,613	7	25	\$3,613
8	23	\$3,613	8	25	\$3,613

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		28			28
1	26	\$3,613	1	27	\$11,576
2	26	\$3,613	2	27	\$11,576
3	26	\$3,613	3	27	\$11,576
4	26	\$3,613	4	27	\$11,576
5	26	\$3,613	5	27	\$11,576
6	26	\$3,613	6	27	\$11,576
7	26	\$3,613	7	27	\$11,576
8	26	\$3,613	8	27	\$11,576

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance			
		32			34	35	36	37
1	30	\$16,517	1	33	\$7,242	\$176	\$263	\$11,576
2	30	\$16,517	2	33	\$7,242	\$300	\$373	\$11,576
3	30	\$16,517	3	33	\$7,242	\$176	\$263	\$11,576
4	30	\$16,517	4	33	\$7,242	\$300	\$373	\$11,576
5	30	\$16,517	5	33	\$7,242	\$176	\$263	\$11,576
6	30	\$16,517	6	33	\$7,242	\$300	\$373	\$11,576
7	30	\$16,517	7	33	\$7,242	\$176	\$263	\$11,576
8	30	\$16,517	8	33	\$7,242	\$300	\$373	\$11,576

Table G.9 (Continued)

Building Type	Initial Resistance	Final Resistance	
		22	24
1	38	\$5,938	\$6,025
2	38	\$5,908	\$5,980
3	38	\$5,938	\$6,025
4	38	\$5,908	\$5,980
5	38	\$5,938	\$6,025
6	38	\$5,908	\$5,980
7	38	\$5,938	\$6,025
8	38	\$5,908	\$5,980

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		40			42
1	39	\$3,613	1	41	\$3,613
2	39	\$3,613	2	41	\$3,613
3	39	\$3,613	3	41	\$3,613
4	39	\$3,613	4	41	\$3,613
5	39	\$3,613	5	41	\$3,613
6	39	\$3,613	6	41	\$3,613
7	39	\$3,613	7	41	\$3,613
8	39	\$3,613	8	41	\$3,613

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		45			46
1	43	\$3,613	1	44	\$3,613
2	43	\$3,613	2	44	\$3,613
3	43	\$3,613	3	44	\$3,613
4	43	\$3,613	4	44	\$3,613
5	43	\$3,613	5	44	\$3,613
6	43	\$3,613	6	44	\$3,613
7	43	\$3,613	7	44	\$3,613
8	43	\$3,613	8	44	\$3,613

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		46			50
1	45	\$11,576	1	48	\$16,517
2	45	\$11,576	2	48	\$16,517
3	45	\$11,576	3	48	\$16,517
4	45	\$11,576	4	48	\$16,517
5	45	\$11,576	5	48	\$16,517
6	45	\$11,576	6	48	\$16,517
7	45	\$11,576	7	48	\$16,517
8	45	\$11,576	8	48	\$16,517

Table G.9 (Continued)

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		56			58
1	55	\$3,613	1	57	\$3,613
2	55	\$3,613	2	57	\$3,613
3	55	\$3,613	3	57	\$3,613
4	55	\$3,613	4	57	\$3,613
5	55	\$3,613	5	57	\$3,613
6	55	\$3,613	6	57	\$3,613
7	55	\$3,613	7	57	\$3,613
8	55	\$3,613	8	57	\$3,613

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		61			62
1	59	\$3,613	1	60	\$3,613
2	59	\$3,613	2	60	\$3,613
3	59	\$3,613	3	60	\$3,613
4	59	\$3,613	4	60	\$3,613
5	59	\$3,613	5	60	\$3,613
6	59	\$3,613	6	60	\$3,613
7	59	\$3,613	7	60	\$3,613
8	59	\$3,613	8	60	\$3,613

Building Type	Initial Resistance	Final Resistance	Building Type	Initial Resistance	Final Resistance
		62			66
1	61	\$11,576	1	64	\$16,517
2	61	\$11,576	2	64	\$16,517
3	61	\$11,576	3	64	\$16,517
4	61	\$11,576	4	64	\$16,517
5	61	\$11,576	5	64	\$16,517
6	61	\$11,576	6	64	\$16,517
7	61	\$11,576	7	64	\$16,517
8	61	\$11,576	8	64	\$16,517



### ***Reconstruction Costs***

Reconstruction costs were estimated for each of the 68 component resistance combinations entering each of the possible 31 damage states. Additionally, the cost to reconstruct the buildings included the cost to rebuild the building to an enhanced building resistance. See Appendix C for a complete description of the simplification to 31 damage states. To estimate the cost associated with each of the damage states, the individual component damage states in Table 3 were translated into descriptions of physical damage and then reconstruction costs to repair the damage were estimated. Tables G.10, G.11, G.12, G.13, G.14, G.15 describe the physical damage requiring repair to for each of the component damage states. Tables G.3, G.4, G.5, G.6, G.7 and G.8 describe the unit costs used for estimation of reconstruction cost.

Table G.10 Repairs associated with roof sheathing component damage states

<b>Roof Sheathing</b>		
<b>DS</b>	<b>Description</b>	<b>% to be Replaced</b>
0	<2% failed	0.00
1	2% failed	0.02
2	5% failed	0.05
3	25% failed (Totaled)	1.00
4	100% failed (Totaled)	1.00

Table G.11 Repairs associated with roof cover component damage states

<b>Roof Cover</b>		
<b>DS</b>	<b>Description</b>	<b>% to be Replaced</b>
0	<2% failed	0.00
1	15% failed	0.15
2	50% failed (Totaled)	1.00
3	100% failed (Totaled)	1.00

Table G.12 Repairs associated with roof-to-wall connection component damage states

Roof-to-Wall Connections		
DS	Description	% to be Replaced
0	No damage	0.00
4	Totaled	1.00

Table G.13 Repairs associated with opening component damage states

Openings			
DS	Description	Windows to be Replaced	Doors to be Replaced
0	No opening failed (window)	0	0
1	1 opening (window)	1	0
2	2 openings (windows)	2	0
3	5 openings (4 windows, 1 door)	4	1
4	13 openings (2 doors, 11 windows)	11	2

Table G.14 Repairs associated with wall component damage states

Walls		
DS	Description	% to be Replaced
0	No damage	0.00
4	Totaled	1.00

Table G.15 Repairs to building associated with flood component damage states

Flood Vulnerability		
DS	Description	% to be Replaced
0	No damage	0.00
1	0-10%	0.05
2	10-20%	0.15
3	20-30%	0.25
4	30-40%	0.35
5	> 40% (Totaled)	1

In the resource allocation model, the reconstruction cost is divided into two portions:

(1) the cost to repair the damage to the original component resistance combination,  $R$ , and (2) the cost to upgrade the building to a higher building resistance level,  $M$ .  $R$  is estimated by summing the cost to repair each of the individual components to the original building resistance level.  $M$  is estimated by summing the costs to repair each of the individual components to the new (mitigated) building resistance level,

including any undamaged component mitigations, and subtracting  $R$  from that value.

Table G.15 enumerates the input values for the cost to repair all 31 damage states,  $R$ .

Table G.16 enumerates the input values for the cost to mitigate while repairing,  $M$ .

## APPENDIX H

### LOSS MODEL MATLAB CODE

#### *fragilitycurve.m*

```
function fragilitycurve(Maxiter)
%Pressure Recalc, no RWC Redistribution
% o = building orientation
% i = iteration number

Mean_Resistance_wood = xlsread('MeanResistance','B16:E25');
% Matrix of mean resistance values at various levels for the building
% components, ..._cmu are resistance capacities for a building with cmu
% walls, % ..._wood are resistances capacities for a building with
% wood walls
Coeff_Variation = xlsread('MeanResistance','F3:F12');
% Array of coefficients of variation for mean resistance values for the
% building components
n = xlsread('MeanResistance','I3:I7');
% Component type: 1 = Roof Cover, 2 = Roof Sheathing, 3 = Roof-to-wall
% connection, 4 = Walls, 5 = Openings
R_CombosG = xlsread('ResistanceCombinations','B3:K21');
R_CombosH = xlsread('ResistanceCombinations','B22:K40');
% Matrix of arrays of resistance level combinations for building
% components
MaxR = xlsread('ResistanceCombinations','M3:M3');
% Number of resistance level components (rows in R_Combos)

Orientation = xlsread('Orientation','B2:B9');
% Array of wind directions (or building orientations to wind)
Model_Building = xlsread('ModelBuilding','D1:K28');
% Matrix containing information about the building and it's geometry -
% Index; # Stories; Wall Material; # Wall-Floor Connections on
% Front/Back; # Wall-Floor Connections on Sides; Length; Width; Mean
% Roof Height; Wall Height; # Windows on Front; # Windows on Back, #
% Windows on Sides, # Windows on Sides in edge pressure region, # Front
% Doors, # Garage Doors, # Back Doors; Roof Shape; Roof Pitch (deg);
% Roof Pitch (rise/run); Roof Height eave to ridge; Roof Width; Hip
% Ridge Length; Angle of Hip Ridge; truss spacing; Sheathing panel
Length; Sheathing
% Panel Width.
w = xlsread('WindSpeeds','A1:A36');
% Array of wind speeds
Maxm = Model_Building(1,end);
% Total number of building types, last Index
global S; %# of stories
global Material; %Wall Material
global WallFloorConFB; %# of WallFloor Connections on Front/Back
global WallFloorConSide; %# of WallFloor Connections on Side
global L; %Length of Building
global W; %Width of Building
global h; %Mean Roof Height
global d; %Wall height
global owf; %# window openings in front
global owb; %# window openings in back
```

```

global owsr; %% window openings on the side
global owse; %% window openings on the side in the edge pressure region
global Lw; %Vertical height of window
global Ww; %Horizontal width of window
global odf; %% front doors
global odg; %% garage doors
global odb; %% back doors
global RoofShape; %Roof shape
global rdeg; %Roof Pitch in Degrees
global r; %Roof Pitch rise/run
global RoofHeight; %Vertical Height of roof eave to ridge
global RoofWidth; %Width of the roof eave to ridge
global HipRidgeLength; %Length of the ridge in the hip connection
global HipRidgeAngle; %Angle of the hip connection
global TrussSpace; %Truss Spacing
global Lp; %Sheathing Panel Length
global Wp; %Sheathing Panel Width
global windspeed; %windspeed for a particular iteration
global panel_areas; %Array of the area of each roof sheathing/cover panel
global panel_coords;
global npanels;
global connection_coords;
global nconnections;

global CCRoofGCp;
CCRoofGCp = xlsread('ExtPressureCoeff','C4:C6');
    %Roof C&C External Pressure Coefficient Values [zone 1; zone 2; zone 3]
global CCWallGCp;
CCWallGCp = xlsread('ExtPressureCoeff','F4:F7');
    %Wall C&C External Pressure Coefficient Values [windward; side leading
    %edge; side; leeward]
global MWFRSCp;
MWFRSCp = xlsread('ExtPressureCoeff','I5:I14');
    %Main Wind Force Resisting System Coefficient Values for zones 1 - 8

DSRC = zeros(100,8,36,19,8);
DSRS = zeros(100,8,36,19,8);
DSRWC = zeros(100,8,36,19,8);
DSW = zeros(100,8,36,19,8);
DSO = zeros(100,8,36,19,8);
DSBuilding = zeros(100,8,36,19,8);

for mindex = 1:Maxm %For each type of building
    m = Model_Building(:,mindex);
    display(mindex);
    building_index = int2str(mindex);
    % attriutes of the current building model of interest
    S = m(2);
    Material = m(3);
    WallFloorConFB = m(4);
    WallFloorConSide = m(5);
    L = m(6);
    W = m(7);
    h = m(8);
    d = m(9);
    owf = m(10);
    owb = m(11);
    owsr = m(12);
    owse = m(13);
    Lw = m(14);
    Ww = m(15);
    odf = m(16);

```

```

odg = m(17);
odb = m(18);
RoofShape = m(19);
rdeg = m(20);
r = m(21);
RoofHeight = m(22);
RoofWidth = m(23);
HipRidgeLength = m(24);
HipRidgeAngle = m(25);
TrussSpace = m(26);
Lp = m(27);
Wp = m(28);

panel_areas = []; %Initialize panel_areas array for building.
panel_coords = []; %Initialize panel_coords array for building.
npanels = []; %Initialize npanels array for building.
connection_coords = []; %Initialize connection_coords array for building.
nconnections = []; %Initialize nconnections array for building.

%Use the correct array of resistances for the wall material
if Material == 1
    Mean_Resistance = Mean_Resistance_wood;
else
    Mean_Resistance = Mean_Resistance_cmu;
end

for Rindex = 1:MaxR % For each combination of component resistances
    % Compile an array, R, of the component resistances
    display(Rindex);
    if RoofShape == 1
        R_Combos = R_CombosH;
    elseif RoofShape == 2
        R_Combos = R_CombosG;
    end
    for index = 1:size(Mean_Resistance,1)
        R(index,1) = Mean_Resistance(index,R_Combos(Rindex,index));
    end
    resistance_index = int2str(Rindex);

    DS0_wind = [];
    DS1_wind = [];
    DS2_wind = [];
    DS3_wind = [];
    DS4_wind = [];

    for windindex = 1:length(w) %For each wind speed
        windspeed = w(windindex);
        display(windspeed);

        DS0 = [];
        DS1 = [];
        DS2 = [];
        DS3 = [];
        DS4 = [];

        Orientation_Results = zeros(length(Orientation),6);
        for oindex = 1:length(Orientation) %For each orientation
            o = Orientation(oindex);
            display(o);

            Iteration_Results = zeros(Maxiter,6);
            for iteration = 1:Maxiter % Run Maxiter iterations

```

```

Preprocess(); %Run Preprocess functions
%Sample component Resistances
[rRC,rRS,rRWC,rW,rO]=SampleResistance(n,R,Coeff_Variation
);

%Calculate Initial Loads
[pRS,pW,pO,pRWC]=InitialLoads(o);

%Initial Failure Check
[rRS,rW,rO,Num_Impact_Failures,rRWCIntermediate]=InitialF
Check(pRS,pW,pO,rRS,rW,rO,o,pRWC,rRWC);

%Recalculate Loads
[pfRS,pfRC,pfRWC,pfW,pfO,pin]=FinalLoads(pRS,rRS,pRWC,pW,
rO,pO,o,rRWC);

%Final Failure Check
[rRC,rRS,rRWC,rW,rO]=FinalFCheck2(o,pin,pfRS,pfRC,pfRWC,p
fW,pfO,rRC,rRS,rRWC,rW,rO);

%Calculate Damage State (individual and building)
[dRC,dRS,dRWC,dW,dO]=ComponentDamageState(rRC,rRS,rRWC,rW
,rO);

[DBuilding]=BuildingDamageState(dRC,dRS,dRWC,dW,dO);

Iteration_Results(iteration,:) =
[dRC,dRS,dRWC,dW,dO,DBuilding];

DSRC(iteration,oindex,windex,Rindex,mindex) = dRC;
DSRS(iteration,oindex,windex,Rindex,mindex) = dRS;
DSRWC(iteration,oindex,windex,Rindex,mindex) = dRWC;
DSW(iteration,oindex,windex,Rindex,mindex) = dW;
DSO(iteration,oindex,windex,Rindex,mindex) = dO;

DSBuilding(iteration,oindex,windex,Rindex,mindex) =
DBuilding;

end

[DS0o,DS1o,DS2o,DS3o,DS4o] =
ProbDamageStates(Iteration_Results, Maxiter);

Orientation_Results(oindex,:) = mean(Iteration_Results);

end
end
end
end

filename = strcat('C:\Documents and Settings\Meredith\My Documents\Research
Project\A-Exam\Matlab Programs\RecentRun\FinalA_VALUES_DamageStates.mat');
save(filename, 'DSRC','DSRS','DSRWC','DSW','DSO','DSBuilding');

```

### ***Preprocess.m***

```
function [] = Preprocess()
```

```
global panel_coords; global npanels; global connection_coords;
global nconnections;
```

```
[panel_coords, npanels] = PPpanelcoords();
[connection_coords, nconnections] = PPrwc_coords();
```

### ***PPpanelcoords.m***

```
function [panel_coords, npanels] = PPpanelcoords()

global RoofShape; %Hip or Gable Roof
global L; %Length of Building
global W; %Width of Building
global RoofWidth; %Width from eave to ridge of side of roof
global HipRidgeAngle; %Angle between eave and roof hip joint ridge
global Lp; %Length of panel
global Wp; %Width of panel

Lr = L; %Length of Building (Same as above, name changed for notation
    purposes)
Sr = W; %Width of Building (Same as above, name changed for notation
    purposes)
Wr = RoofWidth; %Width from eave to ridge of side of roof
    % (Same as above, name changed for notation purposes)
Tr = deg2rad(HipRidgeAngle); %Angle between eave and roof hip joint ridge in
    radians

if RoofShape == 1 % Hip Roof
    %Find coordinates for panels on the front/back of the roof
    j = 1; %Panel number, starting with 1 at the left hand side bottom row
    i = 1; %Panel row, starting with 1 = bottom row

    d = Wp/tan(Tr); %x distance from left base x to left top x
    Lrnew = Lr - d; %x coordinate marking the edge of the side of the roof
        %at the top of the panel

    panel_coords(:,j) = [0;Lp;d;Lp;0;Wp];
    %Dimensions for panel in the bottom left hand corner of side
        %panel_coords(1,j) = left base x coordinate
        %panel_coords(2,j) = right base x coordinate
        %panel_coords(3,j) = left top x coordinate
        %panel_coords(4,j) = right top x coordinate
        %panel_coords(5,j) = base y coordinate
        %panel_coords(6,j) = top y coordinate

    while (panel_coords(4,j) < Lrnew) || (panel_coords(6,j) < Wr)
        %While the current panel is not on the edge of the side,
        %or the panel is not in the top row

        j = j + 1; %advance panel number and find coords of the next panel
        if panel_coords(4,j-1) == Lrnew
            %If the previous panel was at the edge of the row, move up one row
            i = i + 1; %Iterate row number

            if i*d > (i-1)/2*Lp && (i-1)*d < (i-1)/2*Lp
                %Because of the pattern of the panels, if this is true, there
                %will be a fraction of a panel that will fill the space between
                %the edge and the starting point of the next panel, find that
                %fraction-panel's coordinates. The fraction-panel will be a
                %triangular panel.
            end
        end
    end
end
```



```

panel_coords(:,j) = [(i-1)*d; (i-1)/2*Lp; (i-1)/2*Lp; ...
    (i-1)/2*Lp; (i-1)*Wp; ...
    (i-1)*Wp+((i-1)/2*Lp-(i-1)*d)*tan(Tr)];

if panel_coords(6,j) >= Wr
%If that fraction-panel is along the top edge of the side
%then the top coordinate should be the height of the side,
%and the x coordinate on the left top should be modified to
%reflect that the panel may no longer be a triangle, but a
%trapezoid.
    panel_coords(6,j) = Wr;
    panel_coords(3,j) = panel_coords(1,j) + ...
        (Wr-panel_coords(5,j))/tan(Tr);
end

if (panel_coords(6,j) < Wr) || ...
    (panel_coords(3,j) == panel_coords(4,j))
%If the panel is not on the top row or if the panel is a
%triangle, then the coordinates of the next panel are
%subsequently defined
    j = j + 1; %Advance counter
    panel_coords(:,j) = [(i-1)/2*Lp; (i-1)/2*Lp+Lp*i*d; ...
        (i-1)/2*Lp+Lp; (i-1)*Wp; i*Wp];
else
%If the panel was in the top row or is not a triangle, then
%the coordinates of the next panel are subsequently defined
    j = j + 1; %Advance counter
    panel_coords(:,j) = [(i-1)/2*Lp; (i-1)/2*Lp+Lp; ...
        (i-1)/2*Lp; (i-1)/2*Lp+Lp; (i-1)*Wp; i*Wp];
end

else
%If there is no fraction panel at the start of the row, then
%the next panel's coordinates will be as follows
    panel_coords(:,j) = [d*(i-1); (i-1)/2*Lp; d*i; (i-1)/2*Lp; ...
        (i-1)*Wp; i*Wp];

    if panel_coords(6,j) >= Wr
%If that panel is along the top edge of the side
%then the top coordinate should be the height of the side,
%and the x coordinate on the left top should be modified to
%reflect that the panel may no longer be a triangle, but a
%trapezoid.
        panel_coords(6,j) = Wr;
        panel_coords(3,j) = panel_coords(1,j) + ...
            (Wr-panel_coords(5,j))/tan(Tr);
    end

    end

    Lrnew = Lrnew - d; %Calculate the new edge x coordinate by
        %subtracting d from the previous coordinate
else
%If the edge of the current row has not yet been reached, the
%the coordinates of the next panel are as follows
    panel_coords(:,j) = panel_coords(:,j-1)+[Lp;Lp;0;Lp;0;0];
    panel_coords(3,j) = panel_coords(1,j); %adjust dimensions so
    %that panel (which is not on the edge) is a rectangle not
    %trapezoid
end

if panel_coords(4,j) >= Lrnew %If reached right edge of roof adjust
    %The coordinates accordingly to either fill out the edge, or

```

```

%add an additional fraction-panel to fill out the edge.
if panel_coords(4,j) >= Lrnew + d
    %If panel is large enough to fill edge space on its own
    panel_coords(2,j) = Lrnew + d;
    panel_coords(4,j) = Lrnew;
else
    %If panel requires a fraction-panel to fill out the edge.
    panel_coords(4,j) = Lrnew;
    if panel_coords(6,j) >= Wr
        panel_coords(6,j) = Wr;
        panel_coords(3,j) = panel_coords(1,j) + ...
            (Wr-panel_coords(5,j))/tan(Tr);
    end
    %If the edge required a fraction-panel, then the
    %coordinates of that fraction-panel are as follows
    j = j + 1;
    panel_coords(:,j) = [panel_coords(2,j-1);Lrnew+d;...
        panel_coords(2,j-1);panel_coords(2,j-1);...
        panel_coords(5,j-1);panel_coords(5,j-1)+...
        ((Lrnew+d-panel_coords(2,j-1))*tan(Tr))];
end
end

if panel_coords(6,j) >= Wr
    %The top coord cannot be greater than the side's top edge coord
    panel_coords(6,j) = Wr;
end

end

%Adjust the top right coordinate of the final panel
panel_coords(4,end) = panel_coords(2,j) - ...
    (Wr-panel_coords(5,j))/tan(Tr);

panel_coords(:,j+1:2*j) = panel_coords(:,1:j);
%Duplicate the panel coordinates for the identical opposite side of roof
npanels = [j;j]; %j panels front, j panels back

%Find coordinates for panels on the sides of the hip roof
s = 2*j+1; %Total number of panels thus far
i = 1; %Return row iteration to 1

Srnew = Sr - d; %Edge coordinate for top x on current row

panel_coords(:,s) = [0;Lp;d;Lp;0;Wp]; %Bottom left hand panel

check = 0;

%Algorithm the same as for the front/back, see previous comments for
%clarifications
while ((panel_coords(4,s)<Srnew) || (panel_coords(6,s)<Wr)) && check == 0
    s = s + 1;
    if panel_coords(4,s-1) == Srnew
        i = i + 1;
        % First panel in next row
        if i*d > (i-1)/2*Lp && (i-1)*d < (i-1)/2*Lp
            panel_coords(:,s) = [(i-1)*d; (i-1)/2*Lp; (i-1)/2*Lp; ...
                (i-1)/2*Lp; (i-1)*Wp; ...
                (i-1)*Wp+((i-1)/2*Lp-(i-1)*d)*tan(Tr)];
            if panel_coords(6,s) >= Wr
                panel_coords(:,s) = [panel_coords(3,s-1);...
                    panel_coords(4,s-1);panel_coords(3,s-1)+...

```

```

        (Wr-(i-1)*Wp)/tan(Tr);...
        panel_coords(3,s-1)+(Wr-(i-1)*Wp)/tan(Tr);...
        (i-1)*Wp;Wr];
    else
        s = s + 1;
        panel_coords(:,s) = [(i-1)/2*Lp; (i-1)/2*Lp+Lp;i*d;...
            (i-1)/2*Lp+Lp; (i-1)*Wp;i*Wp];
    end
else
    panel_coords(:,s)=[d*(i-1); (i-1)/2*Lp;d*i; (i-1)/2*Lp; (i-1)*Wp;i*Wp];
end
Srnew = Srnew - d;
else
    panel_coords(:,s) = panel_coords(:,s-1)+[Lp;Lp;0;Lp;0;0];
    panel_coords(3,s) = panel_coords(1,s);
    %adjust diagonal to vertical
end

if panel_coords(4,s) >= Srnew %If reached edge of roof
    if panel_coords(4,s) >= Srnew + d
        panel_coords(2,s) = Srnew + d;
        panel_coords(4,s) = Srnew;
    else
        panel_coords(4,s) = Srnew;
        if panel_coords(6,s) >= Wr
            panel_coords(:,s) = [panel_coords(3,s-1);...
                panel_coords(4,s-1);panel_coords(3,s-1)+...
                (Wr-(i-1)*Wp)/tan(Tr);...
                panel_coords(3,s-1)+(Wr-(i-1)*Wp)/tan(Tr);...
                (i-1)*Wp;Wr];
            check = 1;
        else
            s = s + 1;
            panel_coords(:,s) = [panel_coords(2,s-1);Srnew+d;...
                panel_coords(2,s-1);panel_coords(2,s-1);...
                panel_coords(5,s-1);panel_coords(5,s-1)+...
                ((Srnew+d-panel_coords(2,s-1))*tan(Tr))];
        end
    end
end
if panel_coords(6,s) >= Wr
    panel_coords(:,s) = [panel_coords(3,s-1);...
        panel_coords(4,s-1);panel_coords(3,s-1)+...
        (Wr-(i-1)*Wp)/tan(Tr);...
        panel_coords(3,s-1)+(Wr-(i-1)*Wp)/tan(Tr);...
        (i-1)*Wp;Wr];
end

end
panel_coords(:,s+1:(s+(s-2*j))) = panel_coords(:,2*j+1:s);
npanels = [npanels;(s-2*j):(s-2*j)]; %j panels front,back
                                %s-2*j panels on the sides

else %If a gable roof
    j = 1; %Panel Index
    i = 1; %Row Index
    k = 4; %offset value for next row

    panel_coords(:,1) = [0;Lp;0;Wp];

```

```

while (panel_coords(2,j) < Lr) || (panel_coords(4,j) < Wr)
    %While the current panel is not at the right or top edge
    j = j + 1; %iterate panel
    if panel_coords(2,j-1) == Lr
        %If the previous panel was at the right edge of the row
        i = i + 1; %increment row Index
        panel_coords(:,j) = [0;Lp-k;(i-1)*Wp;i*Wp];
        %New panel will start at the left edge of the next row
        if panel_coords(4,j) >= Wr %If reached roof ridge,
            %then highest coord can be roof ridge
            panel_coords(4,j) = Wr;
        end
        j = j + 1;
        %Iterate panel and provide coordinates for next panel in the row

        panel_coords(:,j) = [Lp-k;2*Lp-k;(i-1)*Wp;i*Wp];
        %Change offset value after iterating to a new row, as each row
        %will be offset in an alternating pattern
        if k == 4
            k = 0;
        else
            k = 4;
        end
    end
    else
        %If the previous panel was not at the edge, then the new
        %panel's coordinates are as follows
        panel_coords(:,j) = panel_coords(:,j-1)+[Lp;Lp;0;0];
    end
    if panel_coords(2,j) >= Lr %If reached edge of roof,
        %then the highest x coordinate must be the edge coord
        panel_coords(2,j) = Lr;
    end
    if panel_coords(4,j) >= Wr %If reached roof ridge,
        %then the highest y coordinate must be the edge coord
        panel_coords(4,j) = Wr;
    end
end
end
panel_coords(:,j+1:2*j) = panel_coords(:,1:j); %Duplicate coordinates
npanels = [j;j]; %j panels on front, j panels on back
end

```

### ***PPrwc\_coords.m***

```

function [connection_coords, nconnections]=PPrwc_coords()

global RoofShape; global L; global W;

%Roof-to-Wall connections at 2 ft o.c.
%8 gable end connectors on each end.
%Connections along building length

if RoofShape == 1 %Hip Roof
    con_coords_long = zeros(1,L/2+1);
    i = 1;
    while i < L/2
        i = i + 1;
        con_coords_long(i) = con_coords_long(i-1)+2;
    end
    con_coords_long(end) = L;
    con_coords_short = zeros(1,W/2+1);

```

```

i = 1;
while i < W/2
    i = i + 1;
    con_coords_short(i) = con_coords_short(i-1)+2;
end
con_coords_short(end) = W;
connection_coords = [con_coords_long, con_coords_long, con_coords_short,
con_coords_short];
nconnections = [L/2+1,L/2+1,W/2+1,W/2+1];
else
    con_coords_long = zeros(1,L/2+1);
    i = 1;
    while i < L/2
        i = i + 1;
        con_coords_long(i) = con_coords_long(i-1)+2;
    end
    con_coords_long_gable = con_coords_long(2:end-1);
    gable_coords_left = [0,0,0,0];
    gable_coords_right = [L,L,L,L];
    connection_coords = [gable_coords_left, con_coords_long_gable,
gable_coords_right, gable_coords_left, con_coords_long_gable,
gable_coords_right];
    nconnections = [L/2-1+8,L/2-1+8];
end
end

```

### ***SampleResistance.m***

```

function [rRC,rRS,rRWC,rW,rO]=SampleResistance(n,R,Coeff_Variation)

for nindex = n(1):n(end)
    if nindex == 1
        [rRC]=SampleRC(R(nindex),Coeff_Variation(nindex));
    elseif nindex == 2
        [rRS]=SampleRS(R(nindex),Coeff_Variation(nindex));
    elseif nindex == 3
        [rRWC]=SampleRWCnobatch(R(nindex:nindex+1,1),Coeff_Variation(nindex:
nindex+1,1));
    elseif nindex == 4
        [rW]=SampleW(R(nindex+1:nindex+3,1),Coeff_Variation(nindex+1:nindex+
3,1));
    elseif nindex == 5
        [rO]=SampleO(R(nindex+3:end,1),Coeff_Variation(nindex+3:end,1));
    end;
end;
end;

```

### ***SampleRC.m***

```

function [rRC]=SampleRC(R,cv)
%Sample resistance of each unit of component using the mean resistance and
%the coefficient of variation
%Check that unit resistance is within 2 standard deviations of the mean, if
%not, re-sample the resistance until mean is within 2 standard devaitions

global npanels;
G = sum(npanels);
rRC = (randn(G,1).*cv+ones(G,1)).*R;

for i = 1:G

```

```

        while rRC(i) < R-2*R*cv || rRC(i) > R+2*R*cv
            rRC(i) = (randn*cv+1)*R;
        end;
    end;
end;

```

### ***SampleRS.m***

```

function [rRS]=SampleRS(R,cv)
%Sample resistance of each unit of component using the mean resistance and
%the coefficient of variation
%Check that unit resistance is within 2 standard deviations of the mean, if
%not, re-sample the resistance until mean is within 2 standard deviations

global npanels;
G = sum(npanels);
rRS = (randn(G,1).*cv+ones(G,1)).*R;

for i = 1:G
    while rRS(i) < R-2*R*cv || rRS(i) > R+2*R*cv
        rRS(i) = (randn*cv+1)*R;
    end;
end;
end;

```

### ***SampleRWCnobatch.m***

```

function [rRWC]=SampleRWCnobatch(R,cv)
%Sample resistance of each unit of component using the mean resistance and
%the coefficient of variation
%Check that unit resistance is within 2 standard deviations of the mean, if
%not, re-sample the resistance until mean is within 2 standard deviations

global RoofShape; global nconnections;

Rside = R(1);
Rgable = R(2);
cvside = cv(1);
cvgable = cv(2);

if RoofShape == 1 %Hip Roof
    %Individual Connection Resistances
    rRWC =
    (randn((sum(nconnections)),1).*cvside+ones((sum(nconnections)),1)).*Rside;
    for i = 1:(sum(nconnections))
        while rRWC(i) < Rside-2*Rside*cvside || rRWC(i) >
Rside+2*Rside*cvside
            rRWC(i) = (randn*cvside+1)*Rside;
        end;
    end;
else %Gable Roof
    %Individual Connection Resistances
    rRWC(1:4,1) = (randn(4,1).*cvgable+ones(4,1)).*Rgable;
    rRWC(4+1:nconnections(1)-4,1) =
    (randn(nconnections(1)-8,1).*cvside+ones(nconnections(1)-
8,1)).*Rside;
    rRWC(nconnections(1)-3:nconnections(1),1) =
    (randn(4,1).*cvgable+ones(4,1)).*Rgable;
end;
end;

```

```

rRWC(nconnections(1)+1:nconnections(1)+4,1) =
    (randn(4,1).*cvgable+ones(4,1)).*Rgable;
rRWC(nconnections(1)+5:nconnections(1)+nconnections(2)-4) =
    (randn(nconnections(2)-8,1).*cvside+ones(nconnections(2)-
    8,1)).*Rside;
rRWC(nconnections(1)+nconnections(2)-3:nconnections(1)+nconnections(2)) =
    (randn(4,1).*cvgable+ones(4,1)).*Rgable;
for i = 1:4
    while rRWC(i) < Rgable-2*Rgable*cvgable || rRWC(i) >
Rgable+2*Rgable*cvgable
        rRWC(i) = (randn*cvgable+1)*Rgable;
    end;
end;
for i = 4+1:nconnections(1)-4
    while rRWC(i) < Rside-2*Rside*cvside || rRWC(i) >
Rside+2*Rside*cvside
        rRWC(i) = (randn*cvside+1)*Rside;
    end;
end;
for i = nconnections(1)-3:nconnections(1)
    while rRWC(i) < Rgable-2*Rgable*cvgable || rRWC(i) >
Rgable+2*Rgable*cvgable
        rRWC(i) = (randn*cvgable+1)*Rgable;
    end;
end;
for i = nconnections(1)+1:nconnections(1)+4
    while rRWC(i) < Rgable-2*Rgable*cvgable || rRWC(i) >
Rgable+2*Rgable*cvgable
        rRWC(i) = (randn*cvgable+1)*Rgable;
    end;
end;
for i = nconnections(1)+5:nconnections(1)+nconnections(2)-4
    while rRWC(i) < Rside-2*Rside*cvside || rRWC(i) >
Rside+2*Rside*cvside
        rRWC(i) = (randn*cvside+1)*Rside;
    end;
end;
for i = nconnections(1)+nconnections(2)-3:nconnections(1)+nconnections(2)
    while rRWC(i) < Rgable-2*Rgable*cvgable || rRWC(i) >
Rgable+2*Rgable*cvgable
        rRWC(i) = (randn*cvgable+1)*Rgable;
    end;
end;
end;
end;

```

### ***SampleW.m***

```

function [rW]=SampleW(R,cv)
%Sample resistance of each unit of component using the mean resistance and
%the coefficient of variation
%Check that unit resistance is within 2 standard deviations of the mean, if
%not, re-sample the resistance until mean is within 2 standard deviations

%Roof has three loads - shear, uplift, bending (or out-of-plane)
global Material; global d;

Rshear = R(1,1);
Rup = R(2,1);
Rout = R(3,1);
cvshear = cv(1,1);

```

```

cvup = cv(2,1);
cvout = cv(3,1);

rW(:,1) = (randn(4,1).*cvshear+ones(4,1)).*Rshear;
rW(:,2) = (randn(4,1).*cvup+ones(4,1)).*Rup;
rW(:,3) = (randn(4,1).*cvout+ones(4,1)).*Rout;
for i = 1:4
    while rW(i,1) < Rshear-2*Rshear*cvshear || rW(i,1) >
        Rshear+2*Rshear*cvshear
        rW(i,1) = (randn*cvshear+1)*Rshear;
    end;
    while rW(i,2) < Rup-2*Rup*cvup || rW(i,2) > Rup+2*Rup*cvup
        rW(i,2) = (randn*cvup+1)*Rup;
    end;
    while rW(i,3) < Rout-2*Rout*cvout || rW(i,3) > Rout+2*Rout*cvout
        rW(i,3) = (randn*cvout+1)*Rout;
    end;
end;

if Material == 2
    rW(:,1) = rW(:,1).*(12*12);
    %Units now lb/ft length of the wall (assuming 12" wide wall)
end

```

### ***SampleO.m***

```

function [rO]=SampleO(R,cv)
%Sample resistance of each unit of component using the mean resistance and
%the coefficient of variation
%Check that unit resistance is within 2 standard deviations of the mean, if
%not, re-sample the resistance until mean is within 2 standard deviations

%Three kinds of openings - doors, garage doors, windows.
global owf; %% window openings in front
global owb; %% window openings in back
global owsr; %% window openings on the side
global owse; %% window openings on the side in the edge pressure region
global odf; %% front doors
global odg; %% garage doors
global odb; %% back doors

Rdoor = R(1,1);
RGdoor = R(2,1);
Rwindow = R(3,1);
cvdoor = cv(1,1);
cvGdoor = cv(2,1);
cvwindow = cv(3,1);
Num_door = odf+odb; %Number of Doors (front + back)
Garage = odg; %garage door
Num_windows = owf+owb+2*(owsr+owse); %number of windows (front, back and
sides)

rO(1:Num_door,1) = (randn(Num_door,1).*cvdoor+ones(Num_door,1)).*Rdoor;
rO(Num_door+1:Num_door+Garage,1) =
    (randn(Garage,1).*cvGdoor+ones(Garage,1)).*RGdoor;
rO(Num_door+Garage+1:Num_door+Garage+Num_windows,1) =
    (randn(Num_windows,1).*cvwindow+ones(Num_windows,1)).*Rwindow;

for i = 1:Num_door
    while rO(i) < Rdoor-2*Rdoor*cvdoor || rO(i) > Rdoor+2*Rdoor*cvdoor

```



```

        rO(i) = (randn*cvdoor+1)*Rdoor;
    end;
end;
for i = Num_door+1:Num_door+Garage
    while rO(i) < RGdoor-2*RGdoor*cvGdoor || rO(i) > RGdoor+2*RGdoor*cvGdoor
        rO(i) = (randn*cvGdoor+1)*RGdoor;
    end;
end;
for i = Num_door+Garage+1:Num_door+Garage+Num_windows
    while rO(i) < Rwindow-2*Rwindow*cvwindow || rO(i) >
Rwindow+2*Rwindow*cvwindow
        rO(i) = (randn*cvwindow+1)*Rwindow;
    end;
end;
end;

```

### ***InitialLoads.m***

```

function [pRS,pW,pO,pRWC]=InitialLoads(o)

global windspeed;
G = 0.85;
GCpi = 0.18;
qh = 0.00256*0.85*windspeed^2; %velocity pressure (lb/sqft)

%Roof Sheathing Initial Loads
[pRS,pressure_zone_areas]=ILoadRS(o,GCpi,qh);
[pRWC]=ILoadRWC(pressure_zone_areas,GCpi,qh);
[pW]=ILoadW(G,o,GCpi,qh,pRWC);
[pO]=ILoadO(o,GCpi,qh);

```

### ***ILoadRS.m***

```

function [pRS,pressure_zone_areas]=ILoadRS(o,GCpi,qh)

global panel_areas
global CCRoofGCp;
% CCRoofGCp = Roof zone C&C pressure coefficients [Zone 1; Zone 2; Zone 3]

pz1 = abs(qh*0.8*(CCRoofGCp(1)-GCpi));
pz2 = abs(qh*0.8*(CCRoofGCp(2)-GCpi));
pz3 = abs(qh*0.8*(CCRoofGCp(3)-GCpi));

Pressures = [pz1,pz2,pz3];
[pressure_zone_areas,panel_areas] = preprocessRS(o);
pRS = (Pressures*pressure_zone_areas)';

```

### ***preprocessRS.m***

```

function [pressure_zone_areas,panel_areas] = preprocessRS(o)

%Global Variables
global HipRidgeAngle; %Angle between eave and roof hip joint ridge
global RoofShape; %Hip or Gable Roof
global panel_coords; %Coordinates of sheathing panels

```

```

global npanels; %Number of sheathing panels on each side
[front,back,rside,lside]
Tr = deg2rad(HipRidgeAngle);
%Angle between eave and roof hip joint ridge in radians

if RoofShape == 1 %Hip Roof
    %Calculate the coordinates of the pressure zones given the orientation
    [pzone_coords, nzones] = PPpzonecoords(o);
    panel_areas_matrix = zeros(sum(npanels),4);

    for i = 1:4 %For all four sides
        %Keep track of all panels using index n, count first front panels,
        %then back panels, then right side, then left side.
        if i > 1
            lastpanel = sum(npanels(1:i-1));
        else
            lastpanel = 0;
        end
        for j = 1:npanels(i)
            n = j + lastpanel; %Panel number
            %Panel Coordinates
            x1 = panel_coords(1,n);
            x2 = panel_coords(2,n);
            x3 = panel_coords(3,n);
            x4 = panel_coords(4,n);
            y1 = panel_coords(5,n);
            y2 = panel_coords(6,n);

            %Depending on shape of the panel, divide into M parts, which
            %can more easily be analyzed for their pressure
            if (x1==x3 && x2==x4) || (x2==x3 && x2==x4 && x3==x4) || ...
                (x1==x3 && x1==x4 && x3==x4)
                M = 1;
                Dim(:,1) = [x1;x2;x3;x4;y1;y2];
            elseif x2 == x4 && x1 < x3 && atan((y2-y1)/(x3-x1))-Tr > 0.0005
                M = 3;
                Dim(:,1) = [x1;x2;x1;x2;y1;y2-(x3-x1)*tan(Tr)];
                Dim(:,2) = [x3;x4;x3;x4;y2-(x3-x1)*tan(Tr);y2];
                Dim(:,3) = [x1;x3;x3;x3;y2-(x3-x1)*tan(Tr);y2];
            elseif x1 == x3 && x4 < x2 && atan((y2-y1)/(x2-x4))-Tr > 0.0005
                M = 3;
                Dim(:,1) = [x1;x2;x1;x2;y1;y2-(x2-x4)*tan(Tr)];
                Dim(:,2) = [x1;x3;x1;x3;y2-(x2-x4)*tan(Tr);y2];
                Dim(:,3) = [x3;x4;x1;x1;y2-(x2-x4)*tan(Tr);y2];
            elseif x2 == x4 && x1 < x3 && atan((y2-y1)/(x3-x1))-Tr < 0.0005
                M = 2;
                Dim(:,1) = [x1;x3;x3;x3;y1;y2];
                Dim(:,2) = [x3;x2;x3;x2;y1;y2];
            elseif x1 == x3 && x4 < x2 && atan((y2-y1)/(x2-x4))-Tr < 0.0005
                M = 2;
                Dim(:,1) = [x1;x4;x1;x4;y1;y2];
                Dim(:,2) = [x4;x2;x4;x4;y1;y2];
            elseif x1<x3 && x3<x4 && x4<x2
                if atan((y2-y1)/(x3-x1))-Tr > 0.0005
                    M=4;
                    Dim(:,1) = [x1;x3;x3;x3;y2-tan(Tr)*(x3-x1);y2];
                    Dim(:,2) = [x1;x4;x1;x4;y1;y2-tan(Tr)*(x3-x1)];
                    Dim(:,3) = [x3;x4;x3;x4;y2-tan(Tr)*(x3-x1);y2];
                    Dim(:,4) = [x4;x2;x4;x4;y1;y2];
                elseif atan((y2-y1)/(x2-x4))-Tr > 0.0005
                    M=4;
                    Dim(:,1) = [x1;x3;x3;x3;y1;y2];
                end
            end
        end
    end
end

```

```

        Dim(:,2) = [x3;x4;x3;x4;y2-tan(Tr)*(x2-x4);y2];
        Dim(:,3) = [x3;x2;x3;x2;y1;y2-tan(Tr)*(x2-x4)];
        Dim(:,4) = [x4;x2;x4;x4;y2-tan(Tr)*(x2-x4);y2];
    else
        M=3;
        Dim(:,1) = [x1;x3;x3;x3;y1;y2];
        Dim(:,2) = [x3;x4;x3;x4;y1;y2];
        Dim(:,3) = [x4;x2;x4;x4;y1;y2];
    end
elseif x1 < x3 && x3 == x4 && x4 < x2
    M = 1;
    Dim(:,1) = [x1;x2;x3;x4;y1;y2];
else
    display('Problem with Hip Roof dimensions');
    return;
end

%Initialize the matrix describing the Zone Areas (row 1 =
%pressure zone 1, row 2 = pressure zone 2, row 3 = pressure
%zone 3), for each panel (1..n), defined on each side i. Will
%only fill in those parts of the matrix that are defined.
Az(1:3,n,i) = [0;0;0];

A = [];
for m = 1:M %for all M parts of a panel, calculate area
    mx1 = Dim(1,m);
    mx2 = Dim(2,m);
    mx3 = Dim(3,m);
    mx4 = Dim(4,m);
    my1 = Dim(5,m);
    my2 = Dim(6,m);
    A(m) = 1/2*((mx2-mx1)+(mx4-mx3))*(my2-my1);

    panel_areas_matrix(n,i) = panel_areas_matrix(n,i)+A(m);
    %For all zones on side i, calculate the area of the current
    %portion of the current panel that is in that zone.
    for z = 1:nzones(i)
        zone = pzone_coords(1,z,i);
        Px1 = pzone_coords(3,z,i);
        Px2 = pzone_coords(4,z,i);
        Px3 = pzone_coords(5,z,i);
        Px4 = pzone_coords(6,z,i);
        Py1 = pzone_coords(7,z,i);
        Py2 = pzone_coords(8,z,i);

        if Px2 < mx1 && Px4 < mx3
            area = 0;
        elseif Px1 > mx2 && Px3 - mx4 > 0.00001
            area = 0;
        elseif Py2 <= my1 || Py1 >= my2
            area = 0;
        elseif Px1<=mx1 && Px2>=mx2 && Px3<=mx3 && Px4>=mx4 ...
            && ((Py1<=my1&&Py2>=my1) || (Py1<=my2&&Py2>=my2)) ...
            && mx2 == mx4 && mx1 == mx3
            area = 1/2*((mx2-mx1)+(mx4-mx3))*...
                min((my2-my1),min((my2-Py1),(Py2-my1)));
        elseif Py1 < my2 && Py2 > my1
            if Px1 < Px3
                indicator1 = 1;
            elseif Px1 == Px3
                indicator1 = 0;
            else

```

```

        indicator1 = -1;
    end
    if Px2 < Px4
        indicator2 = 1;
    elseif Px2 == Px4
        indicator2 = 0;
    else
        indicator2 = -1;
    end
    if Py1 >= my1
        Ix1 = max(Px1);
        Ix2 = max(Px2);
    else
        Ix1 = Px1 + (my1-Py1)/tan(Tr)*(indicator1);
        Ix2 = Px2 + (my1-Py1)/tan(Tr)*(indicator2);
        if Px3 == Px4 && Px1>Px3 && Px2>Px4
            if Ix1 < Px3
                Ix1 = Px3;
            end
        elseif Px3 == Px4 && Px1<Px3 && Px2 < Px4
            if Ix2 > Px4
                Ix2 = Px4;
            end
        end
    end
    if Py2 <= my2
        Ix3 = Px3;
        Ix4 = Px4;
    else
        Ix3 = Px3 - (Py2-my2)/tan(Tr)*(indicator1);
        Ix4 = Px4 - (Py2-my2)/tan(Tr)*(indicator2);
        if Px3 == Px4 && Px1>Px3 && Px2>Px4
            if my2 < Py2-(Px2-Px1)*tan(Tr)
                Ix3 = Ix4-3;
            else
                Ix3 = Px3;
            end
        elseif Px3 == Px4 && Px1<Px3 && Px2 < Px4
            if my2 < Py2-(Px2-Px1)*tan(Tr)
                Ix4 = Ix3+3;
            else
                Ix4 = Px4;
            end
        end
    end
    if mx1 == mx3 && mx2 == mx4
        if Px1 < mx2 && mx2 < Px3 && Ix1 < mx2 && ...
            my2-(my1+(mx2-Ix1)*tan(Tr)) > 0.005 ...
            && Ix3-mx4>0
            Iy1 = max(Py1, my1);
            Iy2 = (mx2 - Px1)*tan(Tr);
            area = 1/2*(mx2-Ix1)*(Iy2-Iy1);
        elseif Px3 <= mx2 && mx2 <= Px1 && ...
            Ix1-mx2 > 0 && mx4-Ix3>0
            Iy1 = Py1+(Px1-mx2)*tan(Tr);
            Iy2 = min(Py2,my2);
            area = 1/2*(mx2-Ix3)*(Iy2-Iy1);
            if Ix3 < mx1
                area = area - 1/2*(mx1-Ix3)^2*tan(Tr);
            end
        elseif Px2 < mx1 && mx1 < Px4 && Ix4 > mx1 ...
            && mx1-Ix2>0

```

```

Iy1 = Py1+(mx1-Px2)*tan(Tr);
Iy2 = min(Py2,my2);
area = 1/2*(Ix4-mx1)*(Iy2-Iy1);
elseif Px4 < mx1 && mx1 < Px2 && ...
    my2-(my1+(Ix2-mx2)*tan(Tr))>0.005 &&...
    mx3-Ix4>0
    Iy1 = max(my1,Py1);
    Iy2 = Py1+(Px2-mx1)*tan(Tr);
    area = 1/2*(Ix2-mx1)*(Iy2-Iy1);
    if Px2 > mx2
        area = area - 1/2*(Ix2-mx2)^2*tan(Tr);
    end
elseif Px1 < mx1 && mx1 < Px3 && ...
    Px2 < mx2 && mx2 < Px4 && Ix4 > mx1 ...
    && mx2-Ix2>=0
    area = 1/2*((mx2-mx1)+(mx4-mx3))*min((my2-
    my1), min((Py2-my1),min((my2-Py1),(Py2-
    Py1))))-1/2*(Ix3-mx3)^2*tan(Tr);

    if (mx2-Ix2)*tan(Tr)-(my2-Py1)<0.005
        area = area -1/2*(mx2-Ix2)^2*tan(Tr);
    else
        area = area -1/2*(Ix4-
        Ix2)^2*tan(Tr)...
        -(mx4-Ix4)*min(my2-my1,min(Py2-
        my1,min(my2-Py1,Py2-Py1)));
    end
elseif Px3 < mx1 && mx1 < Px1 && Px4 < mx2 && ...
    mx2 < Px2 && mx3-Ix3>=0
    area = 1/2*((mx2-mx1)+(mx4-mx3))*min((my2-
    my1), min((Py2-my1),min((my2-Py1),(Py2-
    Py1))))-1/2*(Ix1-mx1)^2*tan(Tr)-1/2*(mx4-
    Ix4)^2*tan(Tr);
elseif Px1 < mx1 && mx1 < Px3 && ...
    my2-(my1+(mx1-Ix1)*tan(Tr))>0.005 && ...
    Ix4 > mx1 && mx1-Ix1>0 && Ix3-mx3>0
    area = 1/2*(max(0,min(mx2-mx1,min(mx2-Ix1,...
    min(Ix2-mx1,Ix2-Ix1))))+max(0,...
    min(mx4-mx3,Ix4-mx3)))*...
    min((my2-my1),min((Py2-my1),...
    min((my2-Py1),(Py2-Py1))))...
    -1/2*(Ix3-mx3)^2*tan(Tr);
elseif Px3 < mx1 && mx1 < Px1 && (mx3-Ix3)>=0
    area = 1/2*(max(0,min(mx2-mx1,Ix2-mx1))+...
    max(0,min(mx4-mx3,min(mx4-Ix3,...
    min(Ix4-mx3,Ix4-Ix3)))))*...
    min((my2-my1),min((Py2-my1),...
    min((my2-Py1),(Py2-Py1))))...
    -1/2*(Ix1-mx1)^2*tan(Tr);
elseif Px2 < mx2 && mx2 < Px4 && Ix4 > mx1 &&...
    mx2-Ix2>=0
    area = 1/2*(max(0,min(mx2-mx1,mx2-Ix1))+...
    max(0,min(mx4-mx3,min(mx4-Ix3,...
    min(Ix4-mx3,Ix4-Ix3)))))*...
    min((my2-my1),min((Py2-my1),...
    min((my2-Py1),(Py2-Py1))))...
    -1/2*(mx2-Ix2)^2*tan(Tr);
elseif Px4 < mx2 && mx2 < Px2 && ...
    my2-(my1+(Ix2-mx2)*tan(Tr))>0.005 ...
    && (mx4-Ix4)>0 && (Ix2-mx2)>0
    area = 1/2*(max(0,min(mx2-mx1,min(mx2-Ix1,...
    min(Ix2-mx1,Ix2-Ix1))))+...

```

```

max(0,min(mx4-mx3,mx4-Ix3)))*...
min((my2-my1),min((Py2-my1),...
min((my2-Py1),(Py2-Py1)))...
-1/2*(mx4-Ix4)^2*tan(Tr);
else
area = 1/2*(max(0,min((mx2-mx1),min((mx2-Ix1),
min((Ix2-mx1),(Ix2-Ix1)))))+max(0,min((mx4-
mx3),min((mx4-Ix3),min((Ix4-mx3),(Ix4-
Ix3))))))*min((my2-my1),min((Py2-
my1),min((my2-Py1),(Py2-Py1))));
end
elseif mx3 == mx4
if Px1<mx2 && mx2<Px3 && Ix1<mx2 && Ix2>mx2
area = 1/2*(mx2-Ix1)^2*tan(Tr);
elseif Px4 < mx1 && mx1 < Px2 && mx2-Ix2>0 ...
&& Ix2-mx1>0
area = 1/2*(Ix2-mx1)^2*tan(Tr);
elseif Px2 < mx2 && mx2 < Px4
area = 1/2*((mx2-mx1)+(mx4-mx3))*(my2-my1)-
1/2*(mx2-Ix2)^2*tan(Tr);
if my2 > Py2
area = area-1/2*(my2-Py2)^2/tan(Tr);
end
elseif Px3< mx1 && mx1 < Px1 && Ix1-mx1>0
area = 1/2*((mx2-mx1)+(mx4-mx3))*(my2-my1)-
1/2*(Ix1-mx1)^2*tan(Tr);
if my2-Py2 > 0
area = area-1/2*(my2-Py2)^2/tan(Tr);
end
elseif Py2 < my2
area = 1/2*((max(0,min((mx2-mx1),...
min((mx2-Ix1),min((Ix2-mx1),...
(Ix2-Ix1))))))*min((my2-my1))-...
1/2*(Py2-my2)^2/tan(Tr);
else
area = 1/2*(max(0,min((mx2-mx1),min((mx2-
Ix1),min((Ix2-mx1),(Ix2-Ix1))))))*
min((my2-my1),min((Py2-my1),min((my2-Py1),
(Py2-Py1))));
end
else
display('Problem with matching areas');
return;
end
end
%For a given zone, sum the areas of the portions of the
%panel that are in that zone. The resulting matrix will
%define the area of each panel that is in each of the
%pressure zones on a given side of the roof.
if zone > 0
Az(zone,n,i) = Az(zone,n,i) + area;
end
end
end

%Turn the areas into percentages of the total area of a panel.
for zone = 1:3
Az(zone,n,i) = Az(zone,n,i)/(sum(A));
end
end
end
%Reallocate the Az matrix into a two dimesional matrix, where each

```

```

%column represents one panel, and the three rows represent the percent
%area of that panel in each of the three pressure zone areas.
pressure_zone_areas = [Az(:,1:npanels(1),1),...
    Az(:,npanels(1)+1:npanels(1)+npanels(2),2),...
    Az(:,npanels(1)+npanels(2)+1:npanels(1)+npanels(2)+npanels(3),3),...
    Az(:,npanels(1)+npanels(2)+npanels(3)+1:end,4)];

panel_areas = [panel_areas_matrix(1:npanels(1),1)',...
    panel_areas_matrix(npanels(1)+1:npanels(1)+npanels(2),2)',...
    panel_areas_matrix(npanels(1)+npanels(2)+1:npanels(1)+npanels(2)+npanels(3),3)',...
    panel_areas_matrix(npanels(1)+npanels(2)+npanels(3)+1:end,4)'];
else
    [pzone_coords, nzones] = PPpzonecoords(o);
    for i = 1:2
        for n = 1:npanels(i)
            x1 = panel_coords(1,n);
            x2 = panel_coords(2,n);
            y1 = panel_coords(3,n);
            y2 = panel_coords(4,n);
            W = y2-y1;
            L = x2-x1;
            A = W*L;
            Az(1:3,n,i) = [0;0;0];
            panel_areas_matrix(1:3,n,i) = [0;0;0];
            for z = 1:nzones(i)
                zone = pzone_coords(1,z,i);
                Px1 = pzone_coords(3,z,i);
                Px2 = pzone_coords(4,z,i);
                Py1 = pzone_coords(5,z,i);
                Py2 = pzone_coords(6,z,i);

                if Py2 >= y2 && Px2 >= x2
                    area = max(0,min(W,y2-Py1))*max(0,min(L,x2-Px1));
                    parea = area/A;
                elseif Py2 <= y2 && Px2 >= x2
                    area = max(0,Py2-y1)*max(0,min(L,x2-Px1));
                    parea = area/A;
                elseif Py2 >= y2 && Px2 <= x2
                    area = max(0,min(W,y2-Py1))*max(0,Px2-x1);
                    parea = area/A;
                else %Py2 <= y2 && Px2 <= x2
                    area = max(0,Py2-y1)*max(0,Px2-x1);
                    parea = area/A;
                end

                Az(zone,n,i) = Az(zone,n,i) + parea;
                panel_areas_matrix(zone,n,i) =
                    panel_areas_matrix(zone,n,i)+area;
            end
        end
    end
    pressure_zone_areas = [Az(:, :, 1), Az(:, :, 2)];
    panel_areas = sum([panel_areas_matrix(:, :, 1), panel_areas_matrix(:, :, 2)]);
end

```

### ***PPpzonecoords.m***

```
function [pzone_coords, nzones] = PPpzonecoords(o)

global RoofShape; global L; global W; global RoofWidth; global HipRidgeAngle;
global h;

Lr = L;
Sr = W;
Wr = RoofWidth;
Tr = deg2rad(HipRidgeAngle);
a = max(max(min(0.10*W,0.4*h),0.04*W),3);
b = a;

if RoofShape == 1 %Hip roof
    if o == 0 || o == 180

        %row 1 = pressure zone #
        %row 2 = index of that pressure zone #
        %rows 4-6 = x1;x2;x3;x4;y1;y2
        pzone_coordsW(:,1) = [1;1;a/tan(Tr);Lr-a/tan(Tr);Sr/2;3*Sr/2;a;Wr];
        pzone_coordsW(:,2) = [2;1;b;Lr-b;b+a/tan(Tr);Lr-b-a/tan(Tr);0;a];
        pzone_coordsW(:,3) = [3;1;0;b;a/tan(Tr);a/tan(Tr)+b;0;a];
        pzone_coordsW(:,4) = [3;2;Lr-b;Lr;Lr-a/tan(Tr)-b;Lr-a/tan(Tr);0;a];

        pzone_coordsL(:,1) = [1;1;0;Lr;Sr/2;3*Sr/2;0;Wr];
        pzone_coordsL(:,2) = [0;0;0;0;0;0;0;0];
        pzone_coordsL(:,3) = [0;0;0;0;0;0;0;0];
        pzone_coordsL(:,4) = [0;0;0;0;0;0;0;0];

        pzone_coordsSl(:,1) = [1;1;0;Sr;Sr/2;Sr/2;0;Wr];
        pzone_coordsSl(:,2) = [0;0;0;0;0;0;0;0];
        pzone_coordsSl(:,3) = [0;0;0;0;0;0;0;0];
        pzone_coordsSl(:,4) = [0;0;0;0;0;0;0;0];

        pzone_coordsSr(:,1) = [1;1;0;Sr;Sr/2;Sr/2;0;Wr];
        pzone_coordsSr(:,2) = [0;0;0;0;0;0;0;0];
        pzone_coordsSr(:,3) = [0;0;0;0;0;0;0;0];
        pzone_coordsSr(:,4) = [0;0;0;0;0;0;0;0];

        if o == 0
            pzone_coords(:, :, 1) = pzone_coordsW;
            pzone_coords(:, :, 2) = pzone_coordsL;
            pzone_coords(:, :, 3) = pzone_coordsSl;
            pzone_coords(:, :, 4) = pzone_coordsSr;

            nzones = [4,1,1,1]; %number of zones front, back, side, side
        else
            pzone_coords(:, :, 1) = pzone_coordsL;
            pzone_coords(:, :, 2) = pzone_coordsW;
            pzone_coords(:, :, 3) = pzone_coordsSr;
            pzone_coords(:, :, 4) = pzone_coordsSl;

            nzones = [1,4,1,1];
        end
    end

elseif o == 90 || o == 270

    pzone_coordsW(:,1) = [1;1;b;Lr;Sr/2+b;3*Sr/2;0;Wr];
```



```

pzone_coordsW(:,2) = [2;1;a/tan(Tr);a/tan(Tr)+b;Sr/2;Sr/2+b;3;Wr];
pzone_coordsW(:,3) = [3;1;0;b;a/tan(Tr);a/tan(Tr)+b;0;a];
pzone_coordsW(:,4) = [0;0;0;0;0;0;0;0];

pzone_coordsL(:,1) = [1;1;0;Lr-b;Sr/2;3*Sr/2-b;0;Wr];
pzone_coordsL(:,2) = [2;1;Lr-a/tan(Tr)-b;Lr-a/tan(Tr);3*Sr/2-
b;3*Sr/2;3;Wr];
pzone_coordsL(:,3) = [3;1;Lr-b;Lr;Lr-a/tan(Tr)-b;Lr-a/tan(Tr);0;a];
pzone_coordsL(:,4) = [0;0;0;0;0;0;0;0];

pzone_coordsS1(:,1) = [1;1;a/tan(Tr);Sr-a/tan(Tr);Sr/2;Sr/2;3;Wr];
pzone_coordsS1(:,2) = [2;1;b;Sr-b;b+a/tan(Tr);Sr-b-a/tan(Tr);0;a];
pzone_coordsS1(:,3) = [3;1;0;b;a/tan(Tr);a/tan(Tr)+b;0;a];
pzone_coordsS1(:,4) = [3;2;Sr-b;Sr;Sr-b-a/tan(Tr);Sr-a/tan(Tr);0;a];

pzone_coordsSr(:,1) = [1;1;0;Sr;Sr/2;Sr/2;0;Wr];
pzone_coordsSr(:,2) = [0;0;0;0;0;0;0;0];
pzone_coordsSr(:,3) = [0;0;0;0;0;0;0;0];
pzone_coordsSr(:,4) = [0;0;0;0;0;0;0;0];

if o == 90
    pzone_coords(:,1) = pzone_coordsW;
    pzone_coords(:,2) = pzone_coordsL;
    pzone_coords(:,3) = pzone_coordsS1;
    pzone_coords(:,4) = pzone_coordsSr;

    nzones = [3,3,4,1];

else
    pzone_coords(:,1) = pzone_coordsL;
    pzone_coords(:,2) = pzone_coordsW;
    pzone_coords(:,3) = pzone_coordsSr;
    pzone_coords(:,4) = pzone_coordsS1;

    nzones = [3,3,1,4]; %number of zones front, back, side, side

end

elseif o == 45 || o == 135 || o == 225 || o == 315

    pzone_coordsW(:,1) = [1;1;a/tan(Tr)+b;Lr-
        a/tan(Tr);Sr/2+b;3*Sr/2;a;Wr];
    pzone_coordsW(:,2) = [2;1;b;Lr;b+a/tan(Tr);Lr-a/tan(Tr);0;a];
    pzone_coordsW(:,3) = [2;2;a/tan(Tr);b+a/tan(Tr);Sr/2;Sr/2+b;a;Wr];
    pzone_coordsW(:,4) = [3;1;0;b;a/tan(Tr);b+a/tan(Tr);0;a];
    pzone_coordsW(:,5) = [0;0;0;0;0;0;0;0];

    pzone_coordsW2(:,1) = [1;1;a/tan(Tr);Lr-a/tan(Tr)-b;Sr/2;3*Sr/2-
        b;a;Wr];
    pzone_coordsW2(:,2) = [2;1;0;Lr-b;a/tan(Tr);Lr-a/tan(Tr)-b;0;a];
    pzone_coordsW2(:,3) = [2;2;Lr-a/tan(Tr)-b;Lr-a/tan(Tr);3*Sr/2-
        b;3*Sr/2;a;Wr];
    pzone_coordsW2(:,4) = [3;1;Lr-b;Lr;Lr-b-a/tan(Tr);Lr-a/tan(Tr);0;a];
    pzone_coordsW2(:,5) = [0;0;0;0;0;0;0;0];

    pzone_coordsL(:,1) = [1;1;0;Lr;Sr/2;3*Sr/2;0;Wr];
    pzone_coordsL(:,2) = [0;0;0;0;0;0;0;0];
    pzone_coordsL(:,3) = [0;0;0;0;0;0;0;0];
    pzone_coordsL(:,4) = [0;0;0;0;0;0;0;0];
    pzone_coordsL(:,5) = [0;0;0;0;0;0;0;0];

    pzone_coordsS1(:,1) = [1;1;0;Sr/2;Sr/2;Sr/2;0;Wr];

```

```

pzone_coordsS1(:,2) = [1;2;Sr/2;Sr-b-a/tan(Tr);Sr/2;Sr/2;a;Wr-
    a*tan(Tr)];
pzone_coordsS1(:,3) = [2;1;Sr/2;Sr-b;Sr/2;Sr-b-a/tan(Tr);0;a];
pzone_coordsS1(:,4) = [2;2;Sr-b-a/tan(Tr);Sr-
    a/tan(Tr);Sr/2;Sr/2;a;Wr];
pzone_coordsS1(:,5) = [3;1;Sr-b;Sr;Sr-b-a/tan(Tr);Sr-a/tan(Tr);0;a];

pzone_coordsS12(:,1) = [1;1;b+a/tan(Tr);Sr/2;Sr/2;Sr/2;a;Wr-
    a*tan(Tr)];
pzone_coordsS12(:,2) = [1;2;Sr/2;Sr;Sr/2;Sr/2;0;Wr];
pzone_coordsS12(:,3) = [2;1;b;Sr/2;b+a/tan(Tr);Sr/2;0;a];
pzone_coordsS12(:,4) = [2;2;a/tan(Tr);b+a/tan(Tr);Sr/2;Sr/2;a;Wr];
pzone_coordsS12(:,5) = [3;1;0;b;a/tan(Tr);b+a/tan(Tr);0;a];

pzone_coordsSr(:,1) = [1;1;0;Sr;Sr/2;Sr/2;0;Wr];
pzone_coordsSr(:,2) = [0;0;0;0;0;0;0;0];
pzone_coordsSr(:,3) = [0;0;0;0;0;0;0;0];
pzone_coordsSr(:,4) = [0;0;0;0;0;0;0;0];
pzone_coordsSr(:,5) = [0;0;0;0;0;0;0;0];

if o == 45
    pzone_coords(:,1) = pzone_coordsW;
    pzone_coords(:,2) = pzone_coordsL;
    pzone_coords(:,3) = pzone_coordsS1;
    pzone_coords(:,4) = pzone_coordsSr;

    nzones = [4;1;5;1]; %number of zones front, back, side, side
elseif o == 135
    pzone_coords(:,1) = pzone_coordsL;
    pzone_coords(:,2) = pzone_coordsW2;
    pzone_coords(:,3) = pzone_coordsS12;
    pzone_coords(:,4) = pzone_coordsSr;

    nzones = [1;4;5;1];
elseif o == 225
    pzone_coords(:,1) = pzone_coordsL;
    pzone_coords(:,2) = pzone_coordsW;
    pzone_coords(:,3) = pzone_coordsSr;
    pzone_coords(:,4) = pzone_coordsS1;

    nzones = [1;4;1;5];
else
    pzone_coords(:,1) = pzone_coordsW2;
    pzone_coords(:,2) = pzone_coordsL;
    pzone_coords(:,3) = pzone_coordsSr;
    pzone_coords(:,4) = pzone_coordsS12;

    nzones = [4;1;1;5];
end

else
    display('error in orientation for RS Panel Loads');
    pzone_coords = [];
    nzones = [];
    return;
end

else %Gable roof
    if o == 0 || o == 180
        %row 1 = pressure zone #
        %row 2 = index of that pressure zone #
        %rows 4-6 = x1;x2;y1;y2

```

```

pzone_coordsW(:,1) = [1;1;b;Lr-b;a;Wr];
pzone_coordsW(:,2) = [2;1;0;b;a;Wr];
pzone_coordsW(:,3) = [2;2;b;Lr-b;0;a];
pzone_coordsW(:,4) = [2;3;Lr-b;Lr;a;Wr];
pzone_coordsW(:,5) = [3;1;0;b;0;a];
pzone_coordsW(:,6) = [3;2;Lr-b;Lr;0;a];

pzone_coordsL(:,1) = [1;1;0;Lr;0;Wr];
pzone_coordsL(:,2) = [0;0;0;0;0;0];
pzone_coordsL(:,3) = [0;0;0;0;0;0];
pzone_coordsL(:,4) = [0;0;0;0;0;0];
pzone_coordsL(:,5) = [0;0;0;0;0;0];
pzone_coordsL(:,6) = [0;0;0;0;0;0];

if o == 0
    pzone_coords(:,1) = pzone_coordsW;
    pzone_coords(:,2) = pzone_coordsL;
    nzones = [6,1]; %number of zones front, back
else
    pzone_coords(:,1) = pzone_coordsL;
    pzone_coords(:,2) = pzone_coordsW;
    nzones = [1,6]; %number of zones front, back
end

elseif o == 90 || o == 270
    pzone_coordsW(:,1) = [1;1;b;1/2*Lr;a;Wr-a];
    pzone_coordsW(:,2) = [1;2;1/2*Lr;Lr;0;Wr];
    pzone_coordsW(:,3) = [2;1;0;b;a;Wr-a];
    pzone_coordsW(:,4) = [2;2;b;1/2*Lr;0;a];
    pzone_coordsW(:,5) = [2;3;b;1/2*Lr;Wr-a;Wr];
    pzone_coordsW(:,6) = [3;1;0;b;0;a];
    pzone_coordsW(:,7) = [3;2;0;b;Wr-a;Wr];

    pzone_coordsL(:,1) = [1;1;0;1/2*Lr;0;Wr];
    pzone_coordsL(:,2) = [1;2;1/2*Lr;Lr-b;a;Wr-a];
    pzone_coordsL(:,3) = [2;1;1/2*Lr;Lr-b;0;a];
    pzone_coordsL(:,4) = [2;2;1/2*Lr;Lr-b;Wr-a;Wr];
    pzone_coordsL(:,5) = [2;3;Lr-a;Lr;a;Wr-a];
    pzone_coordsL(:,6) = [3;1;Lr-b;Lr;0;a];
    pzone_coordsL(:,7) = [3;2;Lr-b;Lr;Wr-a;Wr];

    if o == 90
        pzone_coords(:,1) = pzone_coordsW;
        pzone_coords(:,2) = pzone_coordsL;
        nzones = [7,7]; %number of zones front, back
    else
        pzone_coords(:,1) = pzone_coordsL;
        pzone_coords(:,2) = pzone_coordsW;
        nzones = [7,7]; %number of zones front, back
    end

elseif o == 45 || o == 135 || o == 225 || o == 315
    pzone_coordsW(:,1) = [1;1;2*b;Lr;2*a;Wr];
    pzone_coordsW(:,2) = [2;1;2*b;Lr;0;2*a];
    pzone_coordsW(:,3) = [2;2;0;2*b;2*a;Wr];
    pzone_coordsW(:,4) = [3;1;0;2*b;0;2*a];

    pzone_coordsL(:,1) = [1;1;0;1/2*Lr;0;Wr];
    pzone_coordsL(:,2) = [1;2;1/2*Lr;Lr;0;Wr-a];
    pzone_coordsL(:,3) = [2;1;1/2*Lr;Lr;Wr-a;Wr];
    pzone_coordsL(:,4) = [0;0;0;0;0;0];

```

```

pzone_coordsW2(:,1) = [1;1;0;Lr-2*b;2*a;Wr];
pzone_coordsW2(:,2) = [2;1;0;Lr-2*b;0;2*a];
pzone_coordsW2(:,3) = [2;2;Lr-2*b;Lr;2*a;Wr];
pzone_coordsW2(:,4) = [3;1;Lr-2*b;Lr;0;2*a];

pzone_coordsL2(:,1) = [1;1;0;1/2*Lr;0;Wr-a];
pzone_coordsL2(:,2) = [1;2;1/2*Lr;Lr;0;Wr];
pzone_coordsL2(:,3) = [2;1;0;1/2*Lr;Wr-a;Wr];
pzone_coordsL2(:,4) = [0;0;0;0;0;0];

if o == 45
    pzone_coords(:, :, 1) = pzone_coordsW;
    pzone_coords(:, :, 2) = pzone_coordsL;
    nzones = [4,3];

elseif o == 135
    pzone_coords(:, :, 1) = pzone_coordsL2;
    pzone_coords(:, :, 2) = pzone_coordsW2;
    nzones = [3,4];

elseif o == 225
    pzone_coords(:, :, 1) = pzone_coordsL;
    pzone_coords(:, :, 2) = pzone_coordsW;
    nzones = [3,4]; %number of zones front, back
else
    pzone_coords(:, :, 1) = pzone_coordsW2;
    pzone_coords(:, :, 2) = pzone_coordsL2;
    nzones = [4,3]; %number of zones front, back
end

else
    display('error in orientation for RS Panel Loads');
    pzone_coords = [];
    nzones = [];
    return;
end
end
end

```

### ***ILoadRWC.m***

```

function [pRWC]=ILoadRWC(pressure_zone_areas,GCpi,qh)
%Function calculates the load on the individual roof-to-wall connections.
%CCRooFGCp = Roof zone C&C pressure coefficients [Zone 1; Zone 2; Zone 3]
global CCRooFGCp;

pz1 = abs(qh*0.8*(CCRooFGCp(1)-GCpi))-10;
%Pressure minus 10 lbs/sqft dead load
pz2 = abs(qh*0.8*(CCRooFGCp(2)-GCpi))-10;
pz3 = abs(qh*0.8*(CCRooFGCp(3)-GCpi))-10;

Pressures = [pz1,pz2,pz3];

pRS_RWC = (Pressures*pressure_zone_areas)';
[RWCloads]=PPrwc_loads(pRS_RWC);
pRWC = RWCloads;

```

## ***PPrwc\_loads.m***

```
function [RWCloads]=PPrwc_loads(pRS)

%Global Variables
global RoofShape; global panel_coords; global npanels;
global connection_coords; global nconnections; global HipRidgeAngle;
global L; global W; global RoofWidth;

Tr = deg2rad(HipRidgeAngle); %Hip Roof Angle
C = RoofWidth; %Roof Width from eave to ridge

if RoofShape == 1 %Hip Roof
    for i = 1:length(nconnections) %For all sides
        %lastconnection keeps track of final connection number on previous
        %side
        if i > 1
            lastconnection = sum(nconnections(1:i-1));
        else
            lastconnection = 0;
        end
        for j = 1:nconnections(i) %For all connections on this side
            n = j + lastconnection; %n = Connection number
            location = connection_coords(n);
            %connection_coords(n) = dimension from left to right where
            %connection is located along eave of roof

            %Given the connection_coords, define the tributary area by
            %dimcon(1) = x coordinate left, dimcon(2) = x coordinate right,
            %dimcon(3) = y coordinate left, dimcon(4) = y coordinate right.
            if location == 0 %Special case at left end
                dimcon(1,n) = 0;
                dimcon(2,n) = 1;
                dimcon(3,n) = 0;
                dimcon(4,n) = 1*tan(Tr);
            elseif n == lastconnection + nconnections(i)
                %Special case at right end
                dimcon(1,n) = location-1;
                dimcon(2,n) = location;
                dimcon(3,n) = 1*tan(Tr);
                dimcon(4,n) = 0;
            else
                if nconnections(i) == L/2+1 %If on front or back
                    % x coordinates - two foot wide trib area
                    dimcon(1,n) = location-1;
                    dimcon(2,n) = location+1;
                    % y coordinate on left
                    if location-1>=W/2 && location-1<=3*W/2
                        dimcon(3,n) = C; %Connection along ridge
                    elseif location-1<W/2
                        dimcon(3,n) = (location-1)*tan(Tr); %Upslope to ridge
                    else
                        dimcon(3,n) = (connection_coords(nconnections(i))...
                            -location+1)*tan(Tr);
                        %Downslope from ridge
                    end
                    % y coordinate on right
                    if location+1>=W/2 && location+1<=3*W/2
                        dimcon(4,n) = C; %Connection along ridge
                    elseif location+1<W/2
                        dimcon(4,n) = (location+1)*tan(Tr); %Upslope to ridge
```

```

        else
            dimcon(4,n) = (connection_coords(nconnections(i))...
                -location-1)*tan(Tr);
            %Downslope from ridge
        end
    else %If connection is on a side
        % x coordinates - two foot wide trib area
        dimcon(1,n) = location-1;
        dimcon(2,n) = location+1;
        % y coordinate on left
        if location-1 == W/2 %If left edge is at center
            dimcon(3,n) = C;
        elseif location-1 < W/2 %If left edge is left of center
            dimcon(3,n) = (location-1)*tan(Tr);
        else %If left edge is right of center
            dimcon(3,n) = (connection_coords(nconnections(i))...
                -location+1)*tan(Tr);
        end
        % y coordinate on right
        if location+1 == W/2 %If right edge is at center
            dimcon(4,n) = C;
        elseif location+1 < W/2 %If right edge is left of center
            dimcon(4,n) = (location+1)*tan(Tr);
        else %If right edge is right of center
            dimcon(4,n) = (connection_coords(nconnections(i))...
                -location-1)*tan(Tr);
        end
    end
end
end
end
end

else %Gable Roof Shape
    for i = 1:length(nconnections) %For all sides
        %Keep track of the last connection number on the previous side
        if i > 1
            lastconnection = sum(nconnections(1:i-1));
        else
            lastconnection = 0;
        end
        for j = 1:nconnections(i) %For all connections on this side
            n = j + lastconnection; %Connection number
            location = connection_coords(n); %Call up location
            if location == 0 %For left end
                dimcon(1,n) = 0;
                dimcon(2,n) = 1;
                dimcon(3,n) = C;
                dimcon(4,n) = C;
            elseif location == L %For right end
                dimcon(1,n) = L-1;
                dimcon(2,n) = L;
                dimcon(3,n) = C;
                dimcon(4,n) = C;
            else %For all other connections
                dimcon(1,n) = location-1;
                dimcon(2,n) = location+1;
                dimcon(3,n) = C;
                dimcon(4,n) = C;
            end
        end
    end
end
end
end

```

```

RWCloads = zeros(1,n); %Preallocate Load array

%All RWCloads calculated in lbs (lb/sqft * area in sqft)

for i = 1:length(npanels) %For all sides
    %Keep track of last number connection and panel on previous side
    if i > 1
        lastpanel = sum(npanels(1:i-1));
        lastconnection = sum(nconnections(1:i-1));
    else
        lastpanel = 0;
        lastconnection = 0;
    end

    for k = 1:npanels(i) %for all panels on current side, i
        n = k + lastpanel; %assign number to panel
        %get x coordinates of panel
        x1 = panel_coords(1,n);
        x2 = panel_coords(2,n);
        %Get y coordinates of panel
        if RoofShape == 1 %Hip Roof
            y1 = panel_coords(5,n); %Lower y coord
            y2 = panel_coords(6,n); %Upper y coord
        else %Gable Roof
            y1 = panel_coords(3,n); %Lower y coord
            y2 = panel_coords(4,n); %Upper y coord
        end

        for j = 1:nconnections(i) %For all connections on current side, i
            c = j + lastconnection; %Assign number to connection
            %get connection coordinates
            Cx1 = dimcon(1,c);
            Cx2 = dimcon(2,c);
            Cy1 = dimcon(3,c);
            Cy2 = dimcon(4,c);

            if RoofShape == 2 %GABLE ROOF
                %If a part of the panel is in the tributary area of the
                %connection, then find the area of the panel that is within
                %the tributary area boundaries
                if x1 <= Cx2 && x2 >= Cx1 && y1 < Cy2
                    if Cy2 == Cy1
                        RWCloads(c) = RWCloads(c) + pRS(n)*...
                            (min(x2-x1,min(Cx2-Cx1,min(max(x2-Cx1,0),...
                                max(Cx2-x1,0)))))*(min(y2-y1,Cy2-y1));
                    end
                end
            else %HIP ROOF
                %If a part of the panel is in the tributary area of the
                %connection, then find the area of the panel that is within
                %the tributary area boundaries
                if x1 < Cx2 && x2 > Cx1
                    %Panel is in the x boundaries of connection
                    %If connection is along the ridge line
                    if Cy2 == Cy1
                        RWCloads(c) = RWCloads(c) + pRS(n)*...
                            (min(x2-x1,min(Cx2-Cx1,min(max(x2-Cx1,0),...
                                max(Cx2-x1,0)))))*(min(y2-y1,Cy2-y1));
                    end
                    %If connection is on up or down slope between Cy1 and Cy2
                    elseif (Cy1 < y1 && Cy2 > y1 && Cy2 > y2) || ...

```

```

        (Cy1 > y1 && Cy2 < y1 && Cy1 > y2)
        RWClloads(c) = RWClloads(c) + pRS(n)*...
        1/2*(y2-y1)^2/tan(Tr);

%If connection is on up or down slope and panel is above
%lower y coordinate, and above higher y coord
elseif (Cy1 < y1 && Cy2 > y1 && Cy2 <= y2) || ...
    (Cy1 > y1 && Cy2 < y1 && Cy1 <= y2)
    RWClloads(c) = RWClloads(c) + pRS(n)*...
    1/2*(2*max(0,Cy1-y1)+2*max(0,Cy2-y1)-min((Cx2-x1),
    (x2-Cx1))*tan(Tr))*(min(Cx2-x1,x2-Cx1));

elseif (Cy1 < y2 && Cy2 > y2 && x1 < Cx1+(y2-Cy1)/tan(Tr))
    || (Cy1 > y2 && Cy2 < y2 && x2 > Cx2-(y2-
    Cy2)/tan(Tr))
    RWClloads(c) = RWClloads(c) + pRS(n)*...
    ((min(x2-x1,min(Cx2-Cx1,min(max(x2-Cx1,0),...
    max(Cx2-x1,0)))))*(min(Cy1,Cy2)-y1)+...
    1/2*(y2-min(Cy1,Cy2))^2/tan(Tr)+(y2-min(Cy1,Cy2))*
    ((min(x2-x1,min(Cx2-Cx1,min(max(x2-Cx1,0),...
    max(Cx2-x1,0)))))-(y2-min(Cy1,Cy2))/tan(Tr)));

elseif (Cy1 < y2 && Cy2 > y2 && x1 > Cx1+(y2-Cy1)/tan(Tr))
    || (Cy1 > y2 && Cy2 < y2 && x2 < Cx2-(y2-
    Cy2)/tan(Tr))
    RWClloads(c) = RWClloads(c) + pRS(n)*...
    ((min(x2-x1,min(Cx2-Cx1,min(max(x2-Cx1,0),...
    max(Cx2-x1,0)))))*(y2-y1));

else
    RWClloads(c) = RWClloads(c) + pRS(n)*...
    1/2*(min(y2-y1,max(0,Cy1-y1))+min(y2-y1,max(0,Cy2-
    y1)))*(min(x2-x1,min(Cx2-Cx1,min(max(x2-
    Cx1,0),max(Cx2-x1,0)))));
end
end
end
end
end
end
end

if RoofShape == 2 %Gable Roof
    for i = 1:length(nconnections)
        if i > 1
            lastconnection = sum(nconnections(1:i-1));
        else
            lastconnection = 0;
        end
        for j = 1:nconnections(i)
            c = j + lastconnection; %Connection Number
            if j <= 4 || j >= nconnections(i)-3
                RWClloads(c) = RWClloads(c)/4; %Quarter of End Load
            end
        end
    end
end
end

```



## ***ILoadW.m***

```
function [pW]=ILoadW(G,o,GCpi,qh,pRWC)
% Initial loads on the Walls - loads calculated in shear, uplift and
% bending.
% m = building type
% MWFRSGCp = Wall Main Wind Force Resisting System pressure coefficients by
% zones 1-8 [zone 1; ...; zone 8;].
% CCWallGCp = Wall C&C pressure coefficients [windward; side leading area;
% side, leeward]
% pW = pressure value for each wall. Each row represents one of the four
% walls: Front; Back; Left Side; Right Side. The three columns represent
% the three resistances: Shear, Uplift, Out-of-Plane.

global nconnections; global MWFRSCp; global RoofShape; global Material;
global W; global h; global d; global L;
global WallFloorConFB; global WallFloorConSide;

a = max(max(min(0.10*W,0.4*h),0.04*W),3);

%Shear Loads%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initialize Shear Loads
pwshear = abs(qh*0.8*(G*MWFRSCp(1)-GCpi)); %windward pressure (lb/sqft)
plshear = abs(qh*0.8*(G*MWFRSCp(2)-GCpi)); %leeward pressure (lb/sqft)

%Adjust pressure if wind is in a corner direction
if o==45 || o==135 || o==225 || o==315
    pwshear = 0.85*pwshear;
    plshear = 0.85*plshear;
end

% distributed shear load = uniform distributed load
% = shear pressure * half wall height
dwshear = pwshear*d/2; %(lb/ft)
dlshear = plshear*d/2; %(lb/ft)

if o==0 || o==180
    Left = L;
    Right = 0;
elseif o==90 || o==270
    Left = 0;
    Right = W;
elseif o==45 || o==225
    Left = L;
    Right = W;
else
    Left = L;
    Right = W;
end

%Shear Reactions in lbs
V1 = (dwshear*Left/2 + dlshear*Left/2)/L;
V2 = V1;
V3 = (dwshear*Right/2 + dlshear*Right/2)/W;
V4 = V3;

pW(:,1) = [V3;V4;V1;V2];
%pW = [windward, leeward, left side, right side] or
%      = [windward rt, leeward lft, windward lft, leeward rt]

%Uplift Loads assuming all roof-to-wall connections intact%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

%average uplift = mean(pRWC loads in lbs)
if RoofShape == 1 %Hip Roof
    Uplift1 = sum(pRWC(1:nconnections(1)));
    Uplift2 = sum(pRWC(nconnections(1)+1:nconnections(1)+nconnections(2)));
    Uplift3 = sum(pRWC(nconnections(1)+nconnections(2)+1:...
        nconnections(1)+nconnections(2)+nconnections(3)));
    Uplift4 = sum(pRWC(nconnections(1)+nconnections(2)+nconnections(3)+1:...
        nconnections(1)+nconnections(2)+nconnections(3)+nconnections(4)));
else
    Uplift1 = sum(pRWC(5:nconnections(1)-4));
    Uplift2 = sum(pRWC(nconnections(1)+5:nconnections(1)+nconnections(2)-4));
    Uplift3 = sum([pRWC(1:4),pRWC(nconnections(1)+nconnections(2)-
        3:nconnections(1)+nconnections(2))]);
    Uplift4 = sum(pRWC(nconnections(1)-3:nconnections(1)+4));
end

pW(:,2) = [Uplift1/L;Uplift2/L;Uplift3/W;Uplift4/W]; %(lbs/ft)

%Out-of-Plane Loads%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Assume that all RWC are still in tact for initial loads
% pressure in lb/sqft
pwout = abs(qh*0.8*(G*MWFRSCp(1)-GCpi)); %windward pressure
peout = abs(qh*0.8*(G*MWFRSCp(3)-GCpi)); %pressure on the edge region of the
side
psout = abs(qh*0.8*(G*MWFRSCp(3)-GCpi)); %pressure on the non-edge region of
the side
plout = abs(qh*0.8*(G*MWFRSCp(2)-GCpi)); %leeward pressure

%Adjust pressures for indirect/diagonal wind direction
if o==45 || o==135 || o==225 || o==315
    pwout = 0.85*pwout;
    plout = 0.85*plout;
end

if Material == 2 %Concrete walls
    % bending moment per foot of wall - assume simply supported at floor
    % and roof.
    wbend = pwout*d^2/8; %UNITS: (lb/sqft)*1ft*sqft = ft-lb
    lbend = plout*d^2/8;
    ebend = peout*d^2/8;
    sbend = psout*d^2/8;

    if o == 0
        pW(:,3) = [wbend;lbend;sbend;sbend];
    elseif o == 180
        pW(:,3) = [lbend;wbend;sbend;sbend];
    elseif o == 90
        pW(:,3) = [sbend;sbend;wbend;lbend];
    elseif o == 270
        pW(:,3) = [sbend;sbend;lbend;wbend];
    elseif o == 45
        pW(:,3) = [wbend;lbend;wbend;lbend];
    elseif o == 135
        pW(:,3) = [lbend;wbend;wbend;lbend];
    elseif o == 225
        pW(:,3) = [lbend;wbend;lbend;wbend];
    elseif o == 315
        pW(:,3) = [wbend;lbend;lbend;wbend];
    end
else %Wood walls

```

```

%Calculate the lateral force on the floor connections
AreaLongWall = (L-d)*d/2; %units sqft
AreaShortWall = (W-d)*d/2;
if a <= d/2
    AreaEdge = 1/2*a^2;
else
    AreaEdge = 1/2*(d/2)^2 + ((a - d)/2)*(d/2);
end

if o==0 || o==180
    wout = pwout*AreaLongWall;
    lout = plout*AreaLongWall;
    eout = peout*AreaEdge;
    neout = psout*(AreaShortWall - AreaEdge);
    sout = eout + neout;

    wcf = wout/WallFloorConFB; %units: lbs per connection
    lcf = lout/WallFloorConFB;
    scf = sout/WallFloorConSide;

    if o == 0
        pW(:,3) = [wcf;lcf;scf;scf];
    else
        pW(:,3) = [lcf;wcf;scf;scf];
    end

elseif o==90 || o==270
    wout = pwout*AreaShortWall;
    lout = plout*AreaShortWall;
    eout = peout*AreaEdge;
    neout = psout*(AreaLongWall - AreaEdge);
    sout = eout + neout;

    wcf = wout/WallFloorConSide;
    lcf = lout/WallFloorConSide;
    scf = sout/WallFloorConFB;

    if o == 90
        pW(:,3) = [scf;scf;wcf;lcf];
    else
        pW(:,3) = [scf;scf;lcf;wcf];
    end

elseif o==45 || o==215
    woutl = pwout*AreaShortWall;
    woutr = pwout*AreaLongWall;
    loutl = plout*AreaLongWall;
    loutr = plout*AreaShortWall;

    wcf1 = woutl/WallFloorConSide;
    wcfr = woutr/WallFloorConFB;
    lcfl = loutl/WallFloorConFB;
    lcfr = loutr/WallFloorConSide;

    if o == 45
        pW(:,3) = [wcfr;lcfl;wcf1;lcfr];
    else
        pW(:,3) = [lcfl;wcfr;lcfr;wcf1];
    end

else
    woutl = pwout*AreaLongWall;

```

```

woutr = pwout*AreaShortWall;
loutl = plout*AreaShortWall;
loutr = plout*AreaLongWall;

wcfl = woutl/WallFloorConFB;
wcfr = woutr/WallFloorConSide;
lcfl = loutl/WallFloorConSide;
lcfr = loutr/WallFloorConFB;

if o == 135
    pW(:,3) = [lcfr;wcfl;wcfr;lcfl];
else
    pW(:,3) = [wcfl;lcfr;lcfl;wcfr];
end
end
end

```

### ***ILoadO.m***

```

function [pO]=ILoadO(o,GCpi,qh)
% G = number of Openings [doors; garage doors; windows]
% CCWallGCp = Wall C&C pressure coefficients [winward; side leading area;
% side, leeward]
% pO = pressure on the openings [doors; garage door; windows]

global owf; global owb; global owsr; global owse; global odf;
global odg; global odb; global CCWallGCp;

s = owsr + owse; %Total # of side windows

pw = abs(qh*0.8*(CCWallGCp(1)-GCpi));
pe = abs(qh*0.8*(CCWallGCp(2)-GCpi));
ps = abs(qh*0.8*(CCWallGCp(3)-GCpi));
pl = abs(qh*0.8*(CCWallGCp(4)-GCpi));

if o == 0
    frontdoor = pw; %front door
    backdoor = pl; %back door
    garage = pw; %front garage door
    frontwindow = pw; %front windows
    backwindow = pl; %back windows
    sidecenter = ps; %side windows not on edge
    sideedge = pe; %side windows on edge
elseif o == 180
    frontdoor = pl; %front door
    backdoor = pw; %back door
    garage = pl; %front garage door
    frontwindow = pl; %front windows
    backwindow = pw; %back windows
    sidecenter = ps; %side windows not on edge
    sideedge = pe; %side windows on edge
elseif o == 90 || o == 270
    frontdoor = ps; %front door
    backdoor = ps; %back door
    garage = pe; %front garage door
    frontwindow = ps; %front windows
    backwindow = ps; %back windows
    sidewind= pw; %side windows on windward side
    sidelee = pl; %side windows on leeward side

```

```

elseif o == 45 || o == 315
    frontdoor = pw; %front door
    backdoor = pl; %back door
    garage = pw; %front garage door
    frontwindow = pw; %front windows
    backwindow = pl; %back windows
    sidewind = pw; %side on windward corner
    sidelee = pl; %side on leeward corner
elseif o == 135 || o == 225
    frontdoor = pl; %front door
    backdoor = pw; %back door
    garage = pl; %front garage door
    frontwindow = pl; %front windows
    backwindow = pw; %back windows
    sidewind = pw; %side on windward corner
    sidelee = pl; %side on leeward corner
end

p0 = [frontdoor*ones(odf,1);backdoor*ones(odb,1);garage*ones(odg,1)];
p0 = [p0; frontwindow*ones(owf,1); backwindow*ones(owb,1)];

if o == 0 || o == 180
    p0 = [p0; sidecenter*ones(owsr,1); sideedge*ones(owse,1);
        sidecenter*ones(owsr,1); sideedge*ones(owse,1)];
else
    p0 = [p0; sidewind*ones(s,1); sidelee*ones(s,1)];
end

```

### ***InitialFCheck.m***

```

function
[rRS,rW,rO,Num_Impact_Failures,rRWCIntermediate]=InitialFCheck(pRS,pW,pO,rRS,
rW,rO,o,pRWC,rRWC)

[rRS]=IFCRS(pRS,rRS);
[rW]=IFCW(pW,rW);
[rO,Num_Impact_Failures]=IFCO(pO,rO,o);
[rRWCIntermediate]=IFCRWCCheck(pRWC,rRWC);

```

### ***IFCRS.m***

```

function [rRS]=IFCRS(pRS,rRS)
% For all roof sheathing panels, check if load is greater than resistance
% capacity. If resistance capacity is too small, panel fails, return a
% resistance capacity = 0 for that panel.

global npanels;
G = sum(npanels);

for i = 1:G
    if pRS(i) >= rRS(i)
        rRS(i) = 0;
        pRS(i) = 0;
    end
end
end

```

### ***IFCW.m***

```
function [rW]=IFCW(pW,rW)

global Material;
% pW = load on each wall.
% rW = resistance value for each wall. Each row represents one of the four
% walls: Front; Back; Left Side; Right Side. The three columns represent
% the three resistances: Shear, Uplift, Out-of-Plane.

if Material == 2 %Concrete Walls
    %Combined uplift and bending failure U = P/Pallow+M/Mallow
    U = zeros(4,1);
    for i = 1:4
        U(i) = (pW(i,2)/rW(i,2))+(pW(i,3)/rW(i,3));
        if U(i) >= 1
            display('Initial Wall Failure!');
            rW(i,2) = 0; %Mark as failed
            rW(i,3) = 0; %Mark as failed
        end;
    end;
    %Shear
    for i = 1:4
        if pW(i,1) >= rW(i,1)
            display('Initial Wall Failure!');
            rW(i,1) = 0; %Mark as failed
        end;
    end;
else %Wood Walls
    %Uplift
    for i = 1:4
        if pW(i,2) >= rW(i,2)
            rW(i,2) = 0; %Mark as failed
        end;
    end;
    %Bending
    for i = 1:4
        if pW(i,3) >= rW(i,3)
            rW(i,3) = 0; %Mark as failed
        end;
    end;
    %Shear
    for i = 1:4
        if pW(i,1) >= rW(i,1)
            rW(i,1) = 0; %Mark as failed
        end;
    end;
end;
```

### ***IFCO.m***

```
function [rO,Num_Impact_Failures]=IFCO(pO,rO,o)

global owf; global owb; global owsr; global owse; global odf;
global odg; global odb;

Num_doors = odf + odb; %Total number of doors
```

```

Gdoor = odg; %Garage door
Num_window = owf+owb+2*(owsr+owse); %Total number of windows

%Doors
for i = 1:Num_doors
    if pO(i) >= rO(i)
        rO(i) = 0; %mark as failed
    end;
end;
%Garage Door
for i = Num_doors+1:Num_doors+Gdoor
    if pO(i) >= rO(i)
        rO(i) = 0; %mark as failed
    end;
end;
%Windows
%Pressure
for i = Num_doors+Gdoor+1:Num_doors+Gdoor+Num_window
    if pO(i) >= rO(i)
        rO(i) = 0; %mark as failed
    end;
end;

%Impact - for only one size of window currently
[pd]=ImpactSim(o); %outputs probability of damage on front,
    %back and side walls

rOImpact = rand(Num_window,1)*100;
    %resistance of opening to impact

Num_Impact_Failures = 0; %Count of the number of failures due to impact
if o == 0 %Impact on Front Only
    for i = 1:owf
        if rOImpact(i) < pd
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
elseif o == 180 %Impact on Back Only
    for i = owf+1:owf+owb
        if rOImpact(i) < pd
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
elseif o == 90 %Impact on left side only
    for i = owf+owb+1:owf+owb+owsr+owse
        if rOImpact(i) < pd
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
elseif o == 270 %Impact on right side only
    for i = owf+owb+owsr+owse+1:owf+owb+2*(owsr+owse)
        if rOImpact(i) < pd
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
elseif o == 45 %Impact on Front and left side
    for i = 1:owf
        if rOImpact(i) < pd(1)

```

```

        rO(Num_doors+Gdoor+i) = 0; %mark as failed
        Num_Impact_Failures = Num_Impact_Failures + 1;
    end
end
for i = owf+owb+1:owf+owb+owsr+owse
    if rOimpact(i) < pd(2)
        rO(Num_doors+Gdoor+i) = 0; %mark as failed
        Num_Impact_Failures = Num_Impact_Failures + 1;
    end
end
elseif o == 135 %Impact on Back and left side
    for i = owf+1:owf+owb
        if rOimpact(i) < pd(1)
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
    for i = owf+owb+1:owf+owb+owsr+owse
        if rOimpact(i) < pd(2)
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
elseif o == 225 %Impact on Back and right side
    for i = owf+1:owf+owb
        if rOimpact(i) < pd(1)
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
    for i = owf+owb+owsr+owse+1:owf+owb+2*(owsr+owse)
        if rOimpact(i) < pd(2)
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
else % o == 315 %Impact on Front and Right side
    for i = 1:owf
        if rOimpact(i) < pd(1)
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
    for i = owf+owb+owsr+owse+1:owf+owb+2*(owsr+owse)
        if rOimpact(i) < pd(2)
            rO(Num_doors+Gdoor+i) = 0; %mark as failed
            Num_Impact_Failures = Num_Impact_Failures + 1;
        end
    end
end
end
end

```

### ***ImpactSim.m***

```

function [pd]=ImpactSim(o)

global L; global W; global Lw; global Ww; global d; global owf; global owb;
global owsr; global owse; global windspeed;

A = normcdf(windspeed,135,15);

```



```

Na = 100;
B = 0.002*windspeed-0.1;
D = normcdf(windspeed,70,10);

FBArea = L*d;
SideArea = W*d;
WindowArea = Lw*Ww;

w_area_front = (WindowArea*owf)/FBArea;
w_area_back = (WindowArea*owb)/FBArea;
w_area_side = (WindowArea*(owsr+owse))/SideArea;

if o == 0
    C = w_area_front;
elseif o == 180
    C = w_area_back;
elseif o == 90 || o == 270
    C = w_area_side;
elseif o == 45 || o == 315
    C = [w_area_front, w_area_side];
elseif o == 135 || o == 225
    C = [w_area_back, w_area_side];
end

pd = zeros(length(C),1);
for j = 1:length(C)
    pd(j) = 1 - exp(-A*Na*B*C(j)*D);
end

```

### ***IFCRWCCheck.m***

```

function [rRWCIntermediate]=IFCRWCCheck(pRWC,rRWC)

global nconnections;
G = sum(nconnections);
rRWCIntermediate = rRWC;
for i = 1:G
    if pRWC(i) >= rRWC(i)
        rRWCIntermediate(i) = 0;
    end
end
end

```

### ***FinalLoads.m***

```

function
[pfRS,pfRC,pfRWC,pfW,pfO,pin]=FinalLoads(pRS,rRS,pRWC,pW,rO,pO,o,rRWC)

global windspeed;
G = 0.85;
qh = 0.00256*0.85*windspeed^2; %velocity pressure
[pin] = Pressure_Recalc(rO,pO,qh,o);

if pin == 0
    pfO = pO;
    pfRS = pRS;
    pfRWC = pRWC;
    pfW = pW;

```

```

else
    [pfO]=FLoadO(o,qh,pin);
    [pfRS,pressure_zone_areas]=FLoadRS(o,qh,pin);
    [rRS_RWC]=FFCRS(pfRS,rRS);
    [pfRWC]=FLoadRWC(rRS_RWC,pressure_zone_areas,qh,pin);
    [rRWC,RWCFailures,pfRWC]=FFCRWC(pfRWC,rRWC);
    [pfW]=FLoadW(G,o,qh,pfRWC,pin);
end

[pfRC]=FLoadRC(o,qh);

```

### ***Pressure\_Recalc.m***

```

function [pin] = Pressure_Recalc(rO,pO,qh,o)

%New internal pressure is a weighted average of the pressure that caused
%the failure of any openings in the building during the initial failure
%check.

global owf; global owb; global owsr; global owse; global odf;
global odg; global odb;

%If there is a garage door and the door has failed, then gar = 1, otherwise
%gar = 0
if odg == 1 && rO(odf+odb+1) == 0
    gar = 1;
else
    gar = 0;
end

num_windows = owf+owb+2*(owsr+owse);

if o == 0
    for i = 1:odf
        pO(i) = pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = -pO(i);
    end
    pO(odf+odb+odg) = pO(i);
    for i = odf+odb+odg+1:odf+odb+odg+owf
        pO(i) = pO(i);
    end
    for i = odf+odb+odg+owf+1:odf+odb+odg+num_windows
        pO(i) = -pO(i);
    end
elseif o == 180
    for i = 1:odf
        pO(i) = -pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = pO(i);
    end
    pO(odf+odb+odg) = -pO(i);
    for i = odf+odb+odg+1:odf+odb+odg+owf
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+1:odf+odb+odg+owf+owb
        pO(i) = pO(i);
    end

```

```

end
for i = odf+odb+odg+owf+owb:odf+odb+odg+num_windows
    pO(i) = -pO(i);
end
elseif o == 90
    for i = 1:odf
        pO(i) = -pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = -pO(i);
    end
    pO(odf+odb+odg) = -pO(i);
    for i = odf+odb+odg+1:odf+odb+odg+owf+owb
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+owb+1:odf+odb+odg+owf+owb+owsr+owse
        pO(i) = pO(i);
    end
    for i = odf+odb+odg+owf+owb+owsr+owse+1:odf+odb+odg+num_windows
        pO(i) = -pO(i);
    end
elseif o == 270
    for i = 1:odf
        pO(i) = -pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = -pO(i);
    end
    pO(odf+odb+odg) = -pO(i);
    for i = odf+odb+odg+1:odf+odb+odg+owf+owb+owsr+owse
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+owb+owsr+owse+1:odf+odb+odg+num_windows
        pO(i) = pO(i);
    end
elseif o == 45
    for i = 1:odf
        pO(i) = pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = -pO(i);
    end
    pO(odf+odb+odg) = pO(i);
    for i = odf+odb+odg+1:odf+odb+odg+owf
        pO(i) = pO(i);
    end
    for i = odf+odb+odg+owf+1:odf+odb+odg+owf+owb
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+owb+1:odf+odb+odg+owf+owb+owsr+owse
        pO(i) = pO(i);
    end
    for i = odf+odb+odg+owf+owb+owsr+owse+1:odf+odb+odg+num_windows
        pO(i) = -pO(i);
    end
elseif o == 135
    for i = 1:odf
        pO(i) = -pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = pO(i);
    end
end

```

```

pO(odf+odb+odg) = -pO(i);
for i = odf+odb+odg+1:odf+odb+odg+owf
    pO(i) = -pO(i);
end
for i = odf+odb+odg+owf+1:odf+odb+odg+owf+owb
    pO(i) = pO(i);
end
for i = odf+odb+odg+owf+owb+1:odf+odb+odg+owf+owb+owsr+owse
    pO(i) = pO(i);
end
for i = odf+odb+odg+owf+owb+owsr+owse+1:odf+odb+odg+num_windows
    pO(i) = -pO(i);
end
elseif o == 225
    for i = 1:odf
        pO(i) = -pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = pO(i);
    end
    pO(odf+odb+odg) = -pO(i);
    for i = odf+odb+odg+1:odf+odb+odg+owf
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+1:odf+odb+odg+owf+owb
        pO(i) = pO(i);
    end
    for i = odf+odb+odg+owf+owb+1:odf+odb+odg+owf+owb+owsr+owse
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+owb+owsr+owse+1:odf+odb+odg+num_windows
        pO(i) = pO(i);
    end
elseif o == 315
    for i = 1:odf
        pO(i) = pO(i);
    end
    for i = odf+1:odf+odb
        pO(i) = -pO(i);
    end
    pO(odf+odb+odg) = pO(i);
    for i = odf+odb+odg+1:odf+odb+odg+owf
        pO(i) = pO(i);
    end
    for i = odf+odb+odg+owf+1:odf+odb+odg+owf+owb
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+owb+1:odf+odb+odg+owf+owb+owsr+owse
        pO(i) = -pO(i);
    end
    for i = odf+odb+odg+owf+owb+owsr+owse+1:odf+odb+odg+num_windows
        pO(i) = pO(i);
    end
end
end

%For the windows, sum the pressure at the windows that have failed,
%quantify the number of windows that have failed.

failure_pressures = 0;
num_failed_pressures = 0;
for i = odf+odb+odg+1:odf+odb+odg+num_windows
    if rO(i) == 0

```

```

        failure_pressures = failure_pressures + p0(i);
        num_failed_pressures = num_failed_pressures + 1;
    end
end

%For doors, sum the pressure at the doors that have failed, quantify the
%number of doors that have failed.
for i = 1:odf+odb
    if r0(i) == 0
        failure_pressures = failure_pressures + p0(i);
        num_failed_pressures = num_failed_pressures + 1;
    end
end

%The new internal pressure is a weighted average of the pressures at the
%failed garage door, doors, and windows.
if (num_failed_pressures + 4*gar) > 0
    pin=(4*gar*p0(odf+odb+1)+failure_pressures)/(num_failed_pressures+4*gar);
end;

%If the internal pressure has not changed or if no openings have failed,
%then pin is set to 0, otherwise the new internal pressure will be the
%value calculated above. (The initial internal pressure is calculated using
%the equation qh*0.8*(0-.18))
if (num_failed_pressures + gar) == 0 || pin == abs(qh*0.8*(0-0.18))
    pin = 0;
end

```

### ***FLoadO.m***

```

function [pf0]=FLoadO(o,qh,pin)
% CCWallGCp = Wall C&C pressure coefficients [winward; side leading area;
% side, leeward]
% pf0 = pressure on the openings [doors; garage door; windows]

global owf; global owb; global owsr; global owse; global odf; global odg;
global odb; global CCWallGCp;

s = owsr + owse; %Total # of side windows

pw = abs(qh*0.8*(CCWallGCp(1))-pin);
pe = abs(qh*0.8*(CCWallGCp(2))-pin);
ps = abs(qh*0.8*(CCWallGCp(3))-pin);
pl = abs(qh*0.8*(CCWallGCp(4))-pin);

if o == 0
    frontdoor = pw; %front door
    backdoor = pl; %back door
    garage = pw; %front garage door
    frontwindow = pw; %front windows
    backwindow = pl; %back windows
    sidecenter = ps; %side windows not on edge
    sideedge = pe*1/4 + ps*3/4; %side windows on edge
elseif o == 180
    frontdoor = pl; %front door
    backdoor = pw; %back door
    garage = pl; %front garage door
    frontwindow = pl; %front windows
    backwindow = pw; %back windows

```

```

        sidecenter = ps; %side windows not on edge
        sideedge = pe*1/4 + ps*3/4; %side windows on edge
elseif o == 90 || o == 270
    frontdoor = ps; %front door
    backdoor = ps; %back door
    garage = pe*1/8+ps*7/8; %front garage door
    frontwindow = ps; %front windows
    backwindow = ps; %back windows
    sidewind= pw; %side windows on windward side
    sidelee = pl; %side windows on leeward side
elseif o == 45 || o == 315
    frontdoor = pw; %front door
    backdoor = pl; %back door
    garage = pw; %front garage door
    frontwindow = pw; %front windows
    backwindow = pl; %back windows
    sidewind = pw; %side on windward corner
    sidelee = pl; %side on leeward corner
elseif o == 135 || o == 225
    frontdoor = pl; %front door
    backdoor = pw; %back door
    garage = pl; %front garage door
    frontwindow = pl; %front windows
    backwindow = pw; %back windows
    sidewind= pw; %side on windward corner
    sidelee = pl; %side on leeward corner
end

pf0 = [frontdoor*ones(odf,1);backdoor*ones(odb,1);garage*ones(odg,1)];
pf0 = [pf0; frontwindow*ones(owf,1); backwindow*ones(owb,1)];

if o == 0 || o == 180
    pf0 = [pf0; sidecenter*ones(owsr,1); sideedge*ones(owse,1);...
        sidecenter*ones(owsr,1); sideedge*ones(owse,1)];
else
    pf0 = [pf0; sidewind*ones(s,1); sidelee*ones(s,1)];
end

```

## ***FLoadRS.m***

```

function [pfRS,pressure_zone_areas]=FLoadRS(o,qh,pin)

global CCRoofGCp;
% CCRoofGCp = Roof zone C&C pressure coefficients [Zone 1; Zone 2; Zone 3]

pz1 = abs(qh*0.8*(CCRoofGCp(1))-abs(pin));
pz2 = abs(qh*0.8*(CCRoofGCp(2))-abs(pin));
pz3 = abs(qh*0.8*(CCRoofGCp(3))-abs(pin));

Pressures = [pz1,pz2,pz3];
[pressure_zone_areas,panel_areas] = preprocessRS(o);
pfRS = (Pressures*pressure_zone_areas)';

```

### ***FFCRS.m***

```
function [rRS]=FFCRS(pfRS,rRS)

global npanels;
G = sum(npanels);

for i = 1:G
    if pfRS(i) >= rRS(i)
        rRS(i) = 0;
        pfRS(i) = 0;
    end
end
```

### ***FLoadRC.m***

```
function [pfRC]=FLoadRC(o,qh)

global CCRoofGCp;
% CCRoofGCp = Roof zone C&C pressure coefficients [Zone 1; Zone 2; Zone 3]

pz1 = abs(qh*0.8*(CCRoofGCp(1)));
pz2 = abs(qh*0.8*(CCRoofGCp(2)));
pz3 = abs(qh*0.8*(CCRoofGCp(3)));

Pressures = [pz1,pz2,pz3];
[pressure_zone_areas] = preprocessRS(o);
pfRC = (Pressures*pressure_zone_areas)';
```

### ***FLoadRWC.m***

```
function [pfRWC]=FLoadRWC(rRS_RWC,pressure_zone_areas,qh,pin)
%Function calculates the load on the individual roof-to-wall connections.
% CCRoofGCp = Roof zone C&C pressure coefficients [Zone 1; Zone 2; Zone 3]
global CCRoofGCp;

pz1 = (abs(qh*0.8*(CCRoofGCp(1))-pin)-10);
%Pressure minus 10 lbs/sqft dead load
pz2 = (abs(qh*0.8*(CCRoofGCp(2))-pin)-10);
pz3 = (abs(qh*0.8*(CCRoofGCp(3))-pin)-10);

Pressures = [pz1,pz2,pz3];
pfRS_RWC = (Pressures*pressure_zone_areas)';
for i = 1:length(pfRS_RWC)
    if rRS_RWC(i) == 0
        pfRS_RWC(i) = 0;
    end
end

[RWCloads]=PPrwc_loads(pfRS_RWC);
pfRWC = RWCloads;
```

## ***FFCRWC.m***

```
function [rRWC,RWCFailures,pfRWC]=FFCRWC(pfRWC,rRWC)

global nconnections;
global RoofShape;

%Count the number of failed roof connections for reallocation of load
count = 0; %Initialize counter of new failures

%For all connections, check for failure. If failed, advance counter.
for i = 1:sum(nconnections)
    if pfRWC(i) >= rRWC(i)
        rRWC(i) = 0;
        count = count + 1;
    end
end

newfailures = count; %Set number of new failures = counter

while newfailures > 0 %If there was a new failure
    %Distribute load and resistances into new arrays for calculation
    %purposes only. Recompile into original pfRWC and rRWC at end of code.
    if RoofShape == 1 %Hip Roof
        %Connections on Front
        ConFront = rRWC(1:nconnections(1));
        LoadFront = pfRWC(1:nconnections(1));
        %Connections on Back
        ConBack = rRWC((nconnections(1)+1):sum(nconnections(1:2)));
        LoadBack = pfRWC((nconnections(1)+1):sum(nconnections(1:2)));
        %Connections on Right Side
        ConRight = rRWC((sum(nconnections(1:3))+1):sum(nconnections(1:4)));
        LoadRight = pfRWC((sum(nconnections(1:3))+1):sum(nconnections(1:4)));
        %Connections on Left Side
        ConLeft = rRWC((sum(nconnections(1:2))+1):sum(nconnections(1:3)));
        LoadLeft = pfRWC((sum(nconnections(1:2))+1):sum(nconnections(1:3)));
    else
        %Connections on Front
        ConFront = rRWC(5:(nconnections(1)-4));
        LoadFront = pfRWC(5:(nconnections(1)-4));
        %Connections on Back
        ConBack = rRWC((nconnections(1)+5):(sum(nconnections(1:2))-4));
        LoadBack = pfRWC((nconnections(1)+5):(sum(nconnections(1:2))-4));
        %Connections on Right Side
        ConRight = rRWC((nconnections(1)-3):(nconnections(1)+4));
        LoadRight = pfRWC((nconnections(1)-3):(nconnections(1)+4));
        %Connections on Left Side
        ConLeft = [rRWC((sum(nconnections(1:2))-
            3):(sum(nconnections(1:2))));rRWC(1:4)];
        LoadLeft = [pfRWC((sum(nconnections(1:2))-
            3):(sum(nconnections(1:2))))];
    end
    %For each side, sum the total resistance capacity left after failed
    %connection's capacities are set to 0.

    frontfail = sum(ConFront);
    backfail = sum(ConBack);
    Lsidefail = sum(ConLeft);
    Rsidefail = sum(ConRight);

    %Stop if one entire side has unzipped from wall, return total failure.
```



```

        if frontfail == 0 || backfail == 0 || Lsidefail == 0 || Rsidefail == 0
            rRWC = zeros(length(rRWC),1);
            break;
        end
    end

%Count failures on each side of the roof
if RoofShape == 1 %Hip Roof
    %Connections on Front
    CapacityFront = rRWC(1:nconnections(1));
    nfront = nconnections(1);
    %Connections on Back
    CapacityBack = rRWC((nconnections(1)+1):sum(nconnections(1:2)));
    nback = nconnections(2);
    %Connections on Right Side
    CapacityRight = rRWC((sum(nconnections(1:3))+1):sum(nconnections(1:4)));
    nright = nconnections(3);
    %Connections on Left Side
    CapacityLeft = rRWC((sum(nconnections(1:2))+1):sum(nconnections(1:3)));
    nleft = nconnections(4);
else
    %Connections on Front
    CapacityFront = rRWC(5:(nconnections(1)-4));
    nfront = (nconnections(1)-8);
    %Connections on Back
    CapacityBack = rRWC((nconnections(1)+5):(sum(nconnections(1:2))-4));
    nback = (nconnections(2)-8);
    %Connections on Right Side
    CapacityRight = rRWC((nconnections(1)-3):(nconnections(1)+4));
    nright = 8;
    %Connections on Left Side
    CapacityLeft = [rRWC((sum(nconnections(1:2))-
        3):(sum(nconnections(1:2))))];rRWC(1:4)];
    nleft = 8;
end

FailCount_Front = 0;
FailCount_Back = 0;
FailCount_Left = 0;
FailCount_Right = 0;

for i = 1:length(CapacityFront)
    if CapacityFront(i) == 0
        FailCount_Front = FailCount_Front + 1;
    end
end
for i = 1:length(CapacityBack)
    if CapacityBack(i) == 0
        FailCount_Back = FailCount_Back + 1;
    end
end
for i = 1:length(CapacityRight)
    if CapacityRight(i) == 0
        FailCount_Right = FailCount_Right + 1;
    end
end
for i = 1:length(CapacityLeft)
    if CapacityLeft(i) == 0
        FailCount_Left = FailCount_Left + 1;
    end
end
end

```

```

RWCFailures = [0,0,0,0];
if FailCount_Front > nfront/2
    RWCFailures(1) = 1;
end
if FailCount_Back > nback/2
    RWCFailures(2) = 1;
end
if FailCount_Left > nleft/2
    RWCFailures(3) = 1;
end
if FailCount_Right > nright/2
    RWCFailures(4) = 1;
end

```

### ***FLoadW.m***

```

function [pfW]=FLoadW(G,o,qh,pfRWC,pin)
% Initial loads on the Walls - loads calculated in shear, uplift and
% bending.
% m = building type
% MWFRSGCp = Wall Main Wind Force Resisting System pressure coefficients by
% zones 1-8 [zone 1; ...; zone 8;].
% CCWallGCp = Wall C&C pressure coefficients [windward; side leading area;
% side, leeward]
% pW = pressure value for each wall. Each row represents one of the four
% walls: Front; Back; Left Side; Right Side. The three columns represent
% the three resistances: Shear, Uplift, Out-of-Plane.

global nconnections; global MWFRSCp; global RoofShape; global Material;
global W; global h; global d; global L;
global WallFloorConFB; global WallFloorConSide;

a = max(max(min(0.10*W,0.4*h),0.04*W),3);

%Shear Loads%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initialize Shear Loads
pwshear = abs(qh*0.8*(G*MWFRSCp(1))-pin);
plshear = abs(qh*0.8*(G*MWFRSCp(2))-pin);

if o==45 || o==135 || o==225 || o==315
    pwshear = 0.85*pwshear;
    plshear = 0.85*plshear;
end

% distributed shear load = uniform distributed load
% = shear pressure * half wall height
dwshear = pwshear*d/2;
dlshear = plshear*d/2;

if o==0 || o==180
    Left = L;
    Right = 0;
elseif o==90 || o==270
    Left = 0;
    Right = W;
elseif o==45 || o==225
    Left = L;
    Right = W;
else

```

```

    Left = L;
    Right = W;
end

%Shear Reactions
V1 = (dwshear*Left/2 + dlshear*Left/2)/L;
V2 = V1;
V3 = (dwshear*Right/2 + dlshear*Right/2)/W;
V4 = V3;

pfW(:,1) = [V3;V4;V1;V2];
%pW = [front or back, either side]

%Uplift Loads assuming all roof-to-wall connections intact%%%%%%%%%
if RoofShape == 1 %Hip Roof
    Uplift1 = sum(pfRWC(1:nconnections(1)));
    Uplift2 = sum(pfRWC(nconnections(1)+1:nconnections(1)+nconnections(2)));
    Uplift3 = sum(pfRWC(nconnections(1)+nconnections(2)+1:...
        nconnections(1)+nconnections(2)+nconnections(3)));
    Uplift4 = sum(pfRWC(nconnections(1)+nconnections(2)+nconnections(3)+1:...
        nconnections(1)+nconnections(2)+nconnections(3)+nconnections(4)));
else
    Uplift1 = sum(pfRWC(5:nconnections(1)-4));
    Uplift2 = sum(pfRWC(nconnections(1)+5:nconnections(1)+nconnections(2)-
        4));
    Uplift3 = sum([pfRWC(1:4),...
        pfRWC(nconnections(1)+nconnections(2)-
        3:nconnections(1)+nconnections(2))]);
    Uplift4 = sum(pfRWC(nconnections(1)-3:nconnections(1)+4));
end

pfW(:,2) = [Uplift1/L;Uplift2/L;Uplift3/W;Uplift4/W]; %(lbs/ft)

%Out-of-Plane Loads%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pressure
pwout = abs(qh*0.8*(G*MWFRSCp(1))-pin); %windward pressure
peout = abs(qh*0.8*(G*MWFRSCp(3))-pin); %pressure on the edge region of the
side
psout = abs(qh*0.8*(G*MWFRSCp(3))-pin); %pressure on the non-edge region of
the side
plout = abs(qh*0.8*(G*MWFRSCp(2))-pin); %leeward pressure

if o==45 || o==135 || o==225 || o==315
    pwout = 0.85*pwout;
    plout = 0.85*plout;
end

if Material == 2 %Concrete walls
    % bending moment per foot of wall - assume simply supported at floor
    % and roof.
    wband = pwout*d^2/8;
    lbend = plout*d^2/8;
    ebend = peout*d^2/8;
    sbend = psout*d^2/8;

    if o == 0
        sideband = (ebend*a+sbend*(W-a))/W;
        pfW(:,3) = [wband;lbend;sideband;sideband];
        %[front; back; Lside; Rside];
    elseif o == 180
        sideband = (ebend*a+sbend*(W-a))/W;

```

```

        pfW(:,3) = [lbend;wbend;sidebend;sidebend];
    elseif o == 90
        sidebend = (ebend*a+sbend*(L-a))/L;
        pfW(:,3) = [sidebend;sidebend;wbend;lbend];
    elseif o == 270
        sidebend = (ebend*a+sbend*(L-a))/L;
        pfW(:,3) = [sidebend;sidebend;lbend;wbend];
    elseif o == 45
        pfW(:,3) = [wbend;lbend;wbend;lbend];
    elseif o == 135
        pfW(:,3) = [lbend;wbend;wbend;lbend];
    elseif o == 225
        pfW(:,3) = [lbend;wbend;lbend;wbend];
    elseif o == 315
        pfW(:,3) = [wbend;lbend;lbend;wbend];
    end

else %Wood walls
    AreaLongWall = (L-d)*d/2;
    AreaShortWall = (W-d)*d/2;
    if a <= d/2
        AreaEdge = 1/2*a^2;
    else
        AreaEdge = 1/2*(d/2)^2 + (a - d/2)*(d/2);
    end

    if o==0 || o==180
        wout = pwout*AreaLongWall;
        lout = plout*AreaLongWall;
        eout = peout*AreaEdge;
        neout = psout*(AreaShortWall - AreaEdge);
        sout = eout + neout;

        wcf = wout/WallFloorConFB;
        lcf = lout/WallFloorConFB;
        scf = sout/WallFloorConSide;

        if o == 0
            pfW(:,3) = [wcf;lcf;scf;scf];
        else
            pfW(:,3) = [lcf;wcf;scf;scf];
        end
    end

elseif o==90 || o==270
    wout = pwout*AreaShortWall;
    lout = plout*AreaShortWall;
    eout = peout*AreaEdge;
    neout = psout*(AreaLongWall - AreaEdge);
    sout = eout + neout;

    wcf = wout/WallFloorConSide;
    lcf = lout/WallFloorConSide;
    scf = sout/WallFloorConFB;

    if o == 90
        pfW(:,3) = [scf;scf;wcf;lcf];
    else
        pfW(:,3) = [scf;scf;lcf;wcf];
    end
end

elseif o==45 || o==215
    woutl = pwout*AreaShortWall;

```

```

        woutr = pwout*AreaLongWall;
        loutl = plout*AreaLongWall;
        loutr = plout*AreaShortWall;

        wcfl = woutl/WallFloorConSide;
        wcfr = woutr/WallFloorConFB;
        lcfl = loutl/WallFloorConFB;
        lcfr = loutr/WallFloorConSide;

        if o == 45
            pfW(:,3) = [wcfr;lcfl;wcfl;lcfr];
        else
            pfW(:,3) = [lcfl;wcfr;lcfr;wcfl];
        end

    else
        woutl = pwout*AreaLongWall;
        woutr = pwout*AreaShortWall;
        loutl = plout*AreaShortWall;
        loutr = plout*AreaLongWall;

        wcfl = woutl/WallFloorConFB;
        wcfr = woutr/WallFloorConSide;
        lcfl = loutl/WallFloorConSide;
        lcfr = loutr/WallFloorConFB;

        if o == 135
            pfW(:,3) = [lcfr;wcfl;wcfr;lcfl];
        else
            pfW(:,3) = [wcfl;lcfr;lcfl;wcfr];
        end

    end
end
end

```

### ***FinalFCheck.m***

```

function
[rRC,rRS,rRWC,rW,rO]=FinalFCheck(o,pin,pfRS,pfRC,pfRWC,pfW,pfO,rRC,rRS,rRWC,r
W,rO)

global windspeed;

[rO]=FFCO(pfO,rO); %Recheck openings
[rRS]=FFCRS(pfRS,rRS); %Recheck Roof Sheathing
[rRC]=FFCRC(pfRC,rRS,rRC); %Check Roof Cover
[rRWC,RWCFailures]=IFCRWCCheck2(pfRWC,rRWC);

%If more than half of the RWC have failed on any side, the wall loads change
if sum(RWCFailures) > 0
    qh = 0.00256*0.85*windspeed^2; %velocity pressure
    [pfW]=FLoadWAdjust(o,qh,pfW,pin,RWCFailures);
end

[rW]=FFCW(pfW,rW); %Recheck Walls

```

### ***FFCO.m***

```
function [rO]=FFCO(pfO,rO)

global owf; global owb; global owsr; global owse; global odf;
global odg; global odb;

Num_doors = odf + odb; %Total number of doors
Gdoor = odg; %Garage door
Num_window = owf+owb+2*(owsr+owse); %Total number or windows

%Doors
for i = 1:Num_doors
    if pfO(i) >= rO(i)
        rO(i) = 0; %mark as failed
    end;
end;
%Garage Door
for i = Num_doors+1:Num_doors+Gdoor
    if pfO(i) >= rO(i)
        rO(i) = 0; %mark as failed
    end;
end;
%Windows
%Pressure
for i = Num_doors+Gdoor+1:Num_doors+Gdoor+Num_window
    if pfO(i) >= rO(i)
        rO(i) = 0; %mark as failed
    end;
end;
```

### ***FFCRS.m***

```
function [rRS]=FFCRS(pfRS,rRS)

global npanels;
G = sum(npanels);

for i = 1:G
    if pfRS(i) >= rRS(i)
        rRS(i) = 0;
        pfRS(i) = 0;
    end
end
```

### ***FFCRC.m***

```
function [rRC]=FFCRC(pfRC,rRS,rRC)

global npanels;
G = sum(npanels);

%Check Sheathing Panel Failures (Cover fails by default)
for i = 1:G
    if rRS(i) == 0
        rRC(i) = 0;
    end
end
```

```

        end
    end

    %Check Roof Cover Panel failures due to uplift loads on cover
    for i = 1:G
        if pfRC(i) >= rRC(i)
            rRC(i) = 0;
        end
    end
end

```

## ***IFCRWCCheck2.m***

```

function [rRWC,RWCFailures]=IFCRWCCheck2(pRWC,rRWC)

global nconnections; global RoofShape;

G = sum(nconnections);
for i = 1:G
    if pRWC(i) >= rRWC(i)
        rRWC(i) = 0;
    end
end

if RoofShape == 1 %Hip Roof
    %Connections on Front
    ConFront = rRWC(1:nconnections(1));
    %Connections on Back
    ConBack = rRWC((nconnections(1)+1):sum(nconnections(1:2)));
    %Connections on Right Side
    ConRight = rRWC((sum(nconnections(1:3))+1):sum(nconnections(1:4)));
    %Connections on Left Side
    ConLeft = rRWC((sum(nconnections(1:2))+1):sum(nconnections(1:3)));
else
    %Connections on Front
    ConFront = rRWC(5:(nconnections(1)-4));
    %Connections on Back
    ConBack = rRWC((nconnections(1)+5):(sum(nconnections(1:2))-4));
    %Connections on Right Side
    ConRight = rRWC((nconnections(1)-3):(nconnections(1)+4));
    %Connections on Left Side
    ConLeft = [rRWC((sum(nconnections(1:2))-
3):(sum(nconnections(1:2))))];rRWC(1:4)];
end
    %For each side, sum the total resistance capacity left after failed
    %connection's capacities are set to 0.

    frontfail = sum(ConFront);
    backfail = sum(ConBack);
    Lsidefail = sum(ConLeft);
    Rsidefail = sum(ConRight);

    %Stop if one entire side has unzipped from wall, return total failure.
    if frontfail == 0 || backfail == 0 || Lsidefail == 0 || Rsidefail == 0
        rRWC = zeros(length(rRWC),1);
    end

    %Count failures on each side of the roof
    if RoofShape == 1 %Hip Roof
        %Connections on Front

```

```

CapacityFront = rRWC(1:nconnections(1));
nfront = nconnections(1);
%Connections on Back
CapacityBack = rRWC((nconnections(1)+1):sum(nconnections(1:2)));
nback = nconnections(2);
%Connections on Right Side
CapacityRight = rRWC((sum(nconnections(1:3))+1):sum(nconnections(1:4)));
nright = nconnections(3);
%Connections on Left Side
CapacityLeft = rRWC((sum(nconnections(1:2))+1):sum(nconnections(1:3)));
nleft = nconnections(4);
else
    %Connections on Front
    CapacityFront = rRWC(5:(nconnections(1)-4));
    nfront = (nconnections(1)-8);
    %Connections on Back
    CapacityBack = rRWC((nconnections(1)+5):(sum(nconnections(1:2))-4));
    nback = (nconnections(2)-8);
    %Connections on Right Side
    CapacityRight = rRWC((nconnections(1)-3):(nconnections(1)+4));
    nright = 8;
    %Connections on Left Side
    CapacityLeft = [rRWC((sum(nconnections(1:2))-
3):(sum(nconnections(1:2))))];rRWC(1:4)];
    nleft = 8;
end

FailCount_Front = 0;
FailCount_Back = 0;
FailCount_Left = 0;
FailCount_Right = 0;

for i = 1:length(CapacityFront)
    if CapacityFront(i) == 0
        FailCount_Front = FailCount_Front + 1;
    end
end
for i = 1:length(CapacityBack)
    if CapacityBack(i) == 0
        FailCount_Back = FailCount_Back + 1;
    end
end
for i = 1:length(CapacityRight)
    if CapacityRight(i) == 0
        FailCount_Right = FailCount_Right + 1;
    end
end
for i = 1:length(CapacityLeft)
    if CapacityLeft(i) == 0
        FailCount_Left = FailCount_Left + 1;
    end
end

RWCFailures = [0,0,0,0];
if FailCount_Front > nfront/2
    RWCFailures(1) = 1;
end
if FailCount_Back > nback/2
    RWCFailures(2) = 1;
end
if FailCount_Left > nleft/2
    RWCFailures(3) = 1;
end

```



```

end
if FailCount_Right > nright/2
    RWCFailures(4) = 1;
end

```

### ***FLoadWAdjust.m***

```

function [pfW]=FLoadWAdjust(o,qh,pfW,pin,RWCFailures)
% Initial loads on the Walls - loads calculated in shear, uplift and
% bending.
% G = []
% m = building type
% MWFRSGCp = Wall Main Wind Force Resisting System pressure coefficients by
% zones 1-8 [zone 1; ...; zone 8;].
% CCWallGCp = Wall C&C pressure coefficients [windward; side leading area;
% side, leeward]
% pW = pressure value for each wall. Each row represents one of the four
% walls: Front; Back; Left Side; Right Side. The three columns represent
% the three resistances: Shear, Uplift, Out-of-Plane.

global MWFRSCp; global Material; global W; global h; global d;
global L; global WallFloorConFB; global WallFloorConSide;

a = max(max(min(0.10*W,0.4*h),0.04*W),3);
G = 0.85;

%Shear Loads%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Shear does not change when roof-to-wall connections are not intact

%Uplift Loads %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Uplift does not change when roof-to-wall connections are not intact

%Out-of-Plane Loads assuming half roof-to-wall connections are broken%%%

if Material == 2 %Concrete walls

    if RWCFailures(1) == 1
        pfW(1,3) = 2.8*pfW(1,3);
    end
    if RWCFailures(2) == 1
        pfW(2,3) = 2.8*pfW(2,3);
    end
    if RWCFailures(3) == 1
        pfW(3,3) = 2.8*pfW(3,3);
    end
    if RWCFailures(4) == 1
        pfW(4,3) = 2.8*pfW(4,3);
    end
end

else %Wood walls
    pwout = abs(qh*0.8*(G*MWFRSCp(1))-pin);
    peout = abs(qh*0.8*(G*MWFRSCp(3))-pin);
    psout = abs(qh*0.8*(G*MWFRSCp(3))-pin);
    plout = abs(qh*0.8*(G*MWFRSCp(2))-pin);

    if o==45 || o==135 || o==225 || o==315
        pwout = 0.85*pwout;
        plout = 0.85*plout;
    end
end

```

```

AreaLongWall = 1/4*L*d;
AreaShortWall = 1/4*W*d;
AreaEdge = 1/4*a^2;

if o==0 || o==180
    wout = pwout*AreaLongWall;
    lout = plout*AreaLongWall;
    eout = peout*AreaEdge;
    neout = psout*(AreaShortWall - AreaEdge);
    sout = eout + neout;

    wcf = wout/WallFloorConFB;
    lcf = lout/WallFloorConFB;
    scf = sout/WallFloorConSide;

    if o == 0
        pfW(:,3) = [wcf;lcf;scf;scf];
    else
        pfW(:,3) = [lcf;wcf;scf;scf];
    end

elseif o==90 || o==270
    wout = pwout*AreaShortWall;
    lout = plout*AreaShortWall;
    eout = peout*AreaEdge;
    neout = psout*(AreaLongWall - AreaEdge);
    sout = eout + neout;

    wcf = wout/WallFloorConSide;
    lcf = lout/WallFloorConSide;
    scf = sout/WallFloorConFB;

    if o == 90
        pfW(:,3) = [scf;scf;wcf;lcf];
    else
        pfW(:,3) = [scf;scf;lcf;wcf];
    end

elseif o==45 || o==215
    woutl = pwout*AreaShortWall;
    woutr = pwout*AreaLongWall;
    loutl = plout*AreaLongWall;
    loutr = plout*AreaShortWall;

    wcfl = woutl/WallFloorConSide;
    wcfr = woutr/WallFloorConFB;
    lcfl = loutl/WallFloorConFB;
    lcfr = loutr/WallFloorConSide;

    if o == 45
        pfW(:,3) = [wcfr;lcfl;wcfl;lcfr];
    else
        pfW(:,3) = [lcfl;wcfr;lcfr;wcfl];
    end

else
    woutl = pwout*AreaLongWall;
    woutr = pwout*AreaShortWall;
    loutl = plout*AreaShortWall;
    loutr = plout*AreaLongWall;

```

```

        wcfl = woutl/WallFloorConFB;
        wcfr = woutr/WallFloorConSide;
        lcfl = loutl/WallFloorConSide;
        lcfr = loutr/WallFloorConFB;

        if o == 135
            pfW(:,3) = [lcfr;wcfl;wcfr;lcfl];
        else
            pfW(:,3) = [wcfl;lcfr;lcfl;wcfr];
        end
    end
end
end

```

### ***FFCW.m***

```

function [rW]=FFCW(pfW,rW)

global Material;
% pW = load on each wall.
% rW = resistance value for each wall. Each row represents one of the four
% walls: Front; Back; Left Side; Right Side. The three columns represent
% the three resistances: Shear, Uplift, Out-of-Plane.

if Material == 2 %Concrete Walls
    %Combined uplift and bending failure  $U = P/P_{allow} + M/M_{allow}$ 
    U = zeros(4,1);
    for i = 1:4
        if rW(i,2) > 0 && rW(i,3) > 0
            U(i) = (pfW(i,2)/rW(i,2))+(pfW(i,3)/rW(i,3));
            if U(i) >= 1
                rW(i,2) = 0; %Mark as failed
                rW(i,3) = 0; %Mark as failed
            end
        end
    end
    %Shear
    for i = 1:4
        if rW(i,1) > 0
            if pfW(i,1) >= rW(i,1)
                rW(i,1) = 0; %Mark as failed
            end
        end
    end
else %Wood Walls
    %Uplift
    for i = 1:4
        if pfW(i,2) >= rW(i,2)
            rW(i,2) = 0; %Mark as failed
        end;
    end;
    %Bending
    for i = 1:4
        if pfW(i,3) >= rW(i,3)
            rW(i,3) = 0; %Mark as failed
        end;
    end;
    %Shear
    for i = 1:4
        if pfW(i,1) >= rW(i,1)

```

```

        rW(i,1) = 0; %Mark as failed
    end;
end;
end;

```

### ***ComponentDamageState.m***

```

function [dRC,dRS,dRWC,dW,dO]=ComponentDamageState(rRC,rRS,rRWC,rW,rO)

global owf; global owb; global owsr; global owse; global odf; global odg;
global odb; global panel_areas;

%Roof Cover Damage State
CountRC = 0;
AreaRC = 0;
TotalAreaRoof = sum(panel_areas);
for i = 1:length(rRC)
    if rRC(i) == 0
        CountRC = CountRC + 1;
        AreaRC = AreaRC + panel_areas(i);
    end
end

DamageRC = AreaRC/TotalAreaRoof;
if DamageRC <= .02
    dRC = 0;
elseif DamageRC > .02 && DamageRC <= .15
    dRC = 1;
elseif DamageRC > .15 && DamageRC <= .50
    dRC = 2;
elseif DamageRC > .50
    dRC = 3;
end

%Roof Sheathing Damage State
CountRS = 0;
AreaRS = 0;
for i = 1:length(rRS)
    if rRS(i) == 0
        CountRS = CountRS + 1;
        AreaRS = AreaRS + panel_areas(i);
    end
end

DamageRS = AreaRS/TotalAreaRoof;
if CountRS >= 1 && CountRS <= 3
    dRS = 2;
elseif CountRS > 3 && DamageRS <= 0.25
    dRS = 3;
elseif DamageRS > 0.25
    dRS = 4;
else
    dRS = 0;
end

%Roof Damage State - total roof failure = damage state 4
if sum(rRWC) == 0
    dRWC = 4;
else

```

```

        dRWC = 0;
    end

    %Openings Damage State
    Num_doors = odf + odb; %Total number of doors
    Gdoor = odg; %Garage door
    Num_window = owf+owb+2*(owsr+owse); %Total number or windows

    %Doors
    CountDoors = 0;
    for i = 1:Num_doors
        if rO(i) == 0
            CountDoors = CountDoors + 1;
        end;
    end;

    %Garage Door
    CountGarage = 0;
    for i = Num_doors+1:Num_doors+Gdoor
        if rO(i) == 0
            CountGarage = CountGarage + 1;
        end;
    end;

    %Windows
    CountWindows = 0;
    for i = Num_doors+Gdoor+1:Num_doors+Gdoor+Num_window
        if rO(i) == 0
            CountWindows = CountWindows + 1;
        end;
    end;

    CountO = (CountDoors+CountGarage+CountWindows);
    if CountO == 0
        dO = 0;
    elseif CountO == 1
        dO = 1;
    elseif CountO > 1 && CountO <= max(.2*length(rO),3)
        dO = 2;
    elseif CountO > max(.2*length(rO),3) && CountO <= .5*length(rO)
        dO = 3;
    elseif CountO > .5*length(rO)
        dO = 4;
    end

    %Walls
    CountW = 0;
    for i = 1:4
        for j = 1:3
            if rW(i,j) == 0
                CountW = CountW + 1;
            end
        end
    end

    if CountW > 0
        dW = 4;
    else
        dW = 0;
    end
end

```

### ***BuildingDamageState.m***

```
function [DBuilding]=BuildingDamageState(dRC,dRS,dRWC,dW,dO)

if dRWC == 4 || dW == 4 || dO == 4 || dRS == 4
    DBuilding = 4;
elseif dRS == 3 || dO == 3 || dRC == 3
    DBuilding = 3;
elseif dRS == 2 || dO == 2 || dRC == 2
    DBuilding = 2;
elseif dO == 1 || dRC == 1
    DBuilding = 1;
elseif dRC == 0 || dO == 0
    DBuilding = 0;
end
```

### ***ProbDamageStates.m***

```
function [DS0,DS1,DS2,DS3,DS4] = ProbDamageStates(Iteration_Results, Maxiter)

dRC = Iteration_Results(:,1);
dRS = Iteration_Results(:,2);
dRWC = Iteration_Results(:,3);
dW = Iteration_Results(:,4);
dO = Iteration_Results(:,5);
DBuilding = Iteration_Results(:,6);

%calculate probability of damage states

pRC_DS0 = 0;
pRC_DS1 = 0;
pRC_DS2 = 0;
pRC_DS3 = 0;
pRC_DS4 = 0;

pRS_DS0 = 0;
pRS_DS1 = 0;
pRS_DS2 = 0;
pRS_DS3 = 0;
pRS_DS4 = 0;

pRWC_DS0 = 0;
pRWC_DS1 = 0;
pRWC_DS2 = 0;
pRWC_DS3 = 0;
pRWC_DS4 = 0;

pW_DS0 = 0;
pW_DS1 = 0;
pW_DS2 = 0;
pW_DS3 = 0;
pW_DS4 = 0;

pO_DS0 = 0;
pO_DS1 = 0;
pO_DS2 = 0;
pO_DS3 = 0;
pO_DS4 = 0;
```

```

pDBuilding_DS0 = 0;
pDBuilding_DS1 = 0;
pDBuilding_DS2 = 0;
pDBuilding_DS3 = 0;
pDBuilding_DS4 = 0;

for results_index = 1:Maxiter
    if dRC(results_index) == 4
        pRC_DS0 = pRC_DS0 + 1;
        pRC_DS1 = pRC_DS1 + 1;
        pRC_DS2 = pRC_DS2 + 1;
        pRC_DS3 = pRC_DS3 + 1;
        pRC_DS4 = pRC_DS4 + 1;
    elseif dRC(results_index) == 3
        pRC_DS0 = pRC_DS0 + 1;
        pRC_DS1 = pRC_DS1 + 1;
        pRC_DS2 = pRC_DS2 + 1;
        pRC_DS3 = pRC_DS3 + 1;
    elseif dRC(results_index) == 2
        pRC_DS0 = pRC_DS0 + 1;
        pRC_DS1 = pRC_DS1 + 1;
        pRC_DS2 = pRC_DS2 + 1;
    elseif dRC(results_index) == 1
        pRC_DS0 = pRC_DS0 + 1;
        pRC_DS1 = pRC_DS1 + 1;
    elseif dRC(results_index) == 0
        pRC_DS0 = pRC_DS0 + 1;
    end

    if dRS(results_index) == 4
        pRS_DS0 = pRS_DS0 + 1;
        pRS_DS1 = pRS_DS1 + 1;
        pRS_DS2 = pRS_DS2 + 1;
        pRS_DS3 = pRS_DS3 + 1;
        pRS_DS4 = pRS_DS4 + 1;
    elseif dRS(results_index) == 3
        pRS_DS0 = pRS_DS0 + 1;
        pRS_DS1 = pRS_DS1 + 1;
        pRS_DS2 = pRS_DS2 + 1;
        pRS_DS3 = pRS_DS3 + 1;
    elseif dRS(results_index) == 2
        pRS_DS0 = pRS_DS0 + 1;
        pRS_DS1 = pRS_DS1 + 1;
        pRS_DS2 = pRS_DS2 + 1;
    elseif dRS(results_index) == 1
        pRS_DS0 = pRS_DS0 + 1;
        pRS_DS1 = pRS_DS1 + 1;
    elseif dRS(results_index) == 0
        pRS_DS0 = pRS_DS0 + 1;
    end

    if dRWC(results_index) == 4
        pRWC_DS0 = pRWC_DS0 + 1;
        pRWC_DS1 = pRWC_DS1 + 1;
        pRWC_DS2 = pRWC_DS2 + 1;
        pRWC_DS3 = pRWC_DS3 + 1;
        pRWC_DS4 = pRWC_DS4 + 1;
    elseif dRWC(results_index) == 3
        pRWC_DS0 = pRWC_DS0 + 1;
        pRWC_DS1 = pRWC_DS1 + 1;
        pRWC_DS2 = pRWC_DS2 + 1;
        pRWC_DS3 = pRWC_DS3 + 1;
    end
end

```

```

elseif dRWC(results_index) == 2
    pRWC_DS0 = pRWC_DS0 + 1;
    pRWC_DS1 = pRWC_DS1 + 1;
    pRWC_DS2 = pRWC_DS2 + 1;
elseif dRWC(results_index) == 1
    pRWC_DS0 = pRWC_DS0 + 1;
    pRWC_DS1 = pRWC_DS1 + 1;
elseif dRWC(results_index) == 0
    pRWC_DS0 = pRWC_DS0 + 1;
end

if dW(results_index) == 4
    pW_DS0 = pW_DS0 + 1;
    pW_DS1 = pW_DS1 + 1;
    pW_DS2 = pW_DS2 + 1;
    pW_DS3 = pW_DS3 + 1;
    pW_DS4 = pW_DS4 + 1;
elseif dW(results_index) == 3
    pW_DS0 = pW_DS0 + 1;
    pW_DS1 = pW_DS1 + 1;
    pW_DS2 = pW_DS2 + 1;
    pW_DS3 = pW_DS3 + 1;
elseif dW(results_index) == 2
    pW_DS0 = pW_DS0 + 1;
    pW_DS1 = pW_DS1 + 1;
    pW_DS2 = pW_DS2 + 1;
elseif dW(results_index) == 1
    pW_DS0 = pW_DS0 + 1;
    pW_DS1 = pW_DS1 + 1;
elseif dW(results_index) == 0
    pW_DS0 = pW_DS0 + 1;
end

if dO(results_index) == 4
    pO_DS0 = pO_DS0 + 1;
    pO_DS1 = pO_DS1 + 1;
    pO_DS2 = pO_DS2 + 1;
    pO_DS3 = pO_DS3 + 1;
    pO_DS4 = pO_DS4 + 1;
elseif dO(results_index) == 3
    pO_DS0 = pO_DS0 + 1;
    pO_DS1 = pO_DS1 + 1;
    pO_DS2 = pO_DS2 + 1;
    pO_DS3 = pO_DS3 + 1;
elseif dO(results_index) == 2
    pO_DS0 = pO_DS0 + 1;
    pO_DS1 = pO_DS1 + 1;
    pO_DS2 = pO_DS2 + 1;
elseif dO(results_index) == 1
    pO_DS0 = pO_DS0 + 1;
    pO_DS1 = pO_DS1 + 1;
elseif dO(results_index) == 0
    pO_DS0 = pO_DS0 + 1;
end

if DBuilding(results_index) == 4
    pDBuilding_DS0 = pDBuilding_DS0 + 1;
    pDBuilding_DS1 = pDBuilding_DS1 + 1;
    pDBuilding_DS2 = pDBuilding_DS2 + 1;
    pDBuilding_DS3 = pDBuilding_DS3 + 1;
    pDBuilding_DS4 = pDBuilding_DS4 + 1;
elseif DBuilding(results_index) == 3

```



```

        pDBuilding_DS0 = pDBuilding_DS0 + 1;
        pDBuilding_DS1 = pDBuilding_DS1 + 1;
        pDBuilding_DS2 = pDBuilding_DS2 + 1;
        pDBuilding_DS3 = pDBuilding_DS3 + 1;
    elseif DBuilding(results_index) == 2
        pDBuilding_DS0 = pDBuilding_DS0 + 1;
        pDBuilding_DS1 = pDBuilding_DS1 + 1;
        pDBuilding_DS2 = pDBuilding_DS2 + 1;
    elseif DBuilding(results_index) == 1
        pDBuilding_DS0 = pDBuilding_DS0 + 1;
        pDBuilding_DS1 = pDBuilding_DS1 + 1;
    elseif DBuilding(results_index) == 0
        pDBuilding_DS0 = pDBuilding_DS0 + 1;
    end
end

DS0 = [pRC_DS0, pRS_DS0, pRWC_DS0, pW_DS0, pO_DS0, pDBuilding_DS0];
DS1 = [pRC_DS1, pRS_DS1, pRWC_DS1, pW_DS1, pO_DS1, pDBuilding_DS1];
DS2 = [pRC_DS2, pRS_DS2, pRWC_DS2, pW_DS2, pO_DS2, pDBuilding_DS2];
DS3 = [pRC_DS3, pRS_DS3, pRWC_DS3, pW_DS3, pO_DS3, pDBuilding_DS3];
DS4 = [pRC_DS4, pRS_DS4, pRWC_DS4, pW_DS4, pO_DS4, pDBuilding_DS4];

DS0 = DS0./Maxiter;
DS1 = DS1./Maxiter;
DS2 = DS2./Maxiter;
DS3 = DS3./Maxiter;
DS4 = DS4./Maxiter;

```

## APPENDIX I

### LOSS MODEL MATLAB USER MANUAL

#### ***Input Data for Loss Model:***

- ModelBuilding.xls - Defines the elements on the example buildings. The default 8 buildings types are the same footprint with one or two stories, hip or gable roof, garage or no garage. All other elements of the building that must be defined to accurately estimate damage are included in the spreadsheet.
- MeanResistance.xls - Defines the mean resistance and coefficient of variation values for the elements of the building for the different configurations and loading types. The default spreadsheet includes strength values for both wood framed buildings and concrete masonry unit buildings.
- Orientation.xls - Defines the wind orientations to the standard building. In the default case, there are 8 wind orientations -  $0^\circ$  (perpendicular to the front of the house),  $45^\circ$  (perpendicular to the corner of the house), and so on at  $45^\circ$  increments around the house.
- ExtPressureCoeff.xls - Defines the external pressure coefficient values ( $GC_p$ ) as defined by ASCE-7 for the wall and roof of the residential structure.
- ResistanceCombinations.xls - Defines the resistance combinations simulated for each of the building types. Each row represents a resistance combination, and each column represents an element. Each entry in the resistance combination defines the mean resistance value for that structural element. For example, a "1" for Roof Cover indicates that the strength of the roof cover will be sampled from the mean resistance 1 (or low) for Roof Cover from the MeanResistance spreadsheet.
- WindSpeeds.xls - Array of the wind speeds for which probabilities of damage are calculated. Default are wind speeds from 50-225 mph at 5 mph increments.

## ***Map of Program:***

### **fragilitycurve.m**

Running fragilitycurve.m results in an output of A, where A is the probability of a building of type m, with resistance combination R, impacted by windspeed w, entering damage state d. Enter Maxiter to define the number of simulation iterations for each resulting A value. The Maxiter used for Meredith Legg's research was 100. fragilitycurve.m begins by loading all of the input data from the excel spreadsheets and assigning the input to global variables. Then the program runs through some preprocessing, and then through the loading/damage check loop for each building type, resistance combination, wind speed and Maxiter simulated buildings.

#### **→Preprocess.m**

Defines the global variables necessary for calculating the roof sheathing panel locations (to determine their loading according to ASCE-7) and the roof to wall connection locations.

##### **→PPpanelcoords.m**

Calculates the coordinates of each of the roof panel corners. These coordinates will be used to calculate the load on each of the panels given the loading pattern of ASCE-7 wind loads.

##### **→PPrwc\_coords.m**

Based on the roof shape, defines the location of the roof to wall connections.

#### **→SampleResistance.m**

Identifies the appropriate mean resistance value based on the resistance combination for each of the structural elements. Runs through the sampling programs for each of the five structural elements.

##### **→SampleRC.m**

Calculate a strength value for each roof cover section from the mean resistance defined by the resistance combination.

##### **→SampleRS.m**

Calculate a strength value for each roof sheathing section from the mean resistance defined by the resistance combination.

##### **→SampleRWCnobatch.m**

Calculate a strength value for each roof to wall connection from the mean resistance defined by the resistance combination.

##### **→SampleW.m**

Calculate a strength value for each wall from the mean resistance defined by the resistance combination.

##### **→SampleO.m**

Calculate a strength value for each window, door and garage door from the mean resistance defined by the resistance combination.

#### **→InitialLoads.m**

Given the ASCE-7 loading coefficients and pressure calculation equations, calculate the initial load for roof sheathing panels, roof to wall connections, walls, and openings.

- **ILoadRS.m**  
Calculate the pressure on each of the roof sheathing panels given the locations of the roof sheathing panels calculated during the preprocess programs.
- **preprocessRS.m**  
Given the orientation of the wind for the particular sampled building, use the location of the sheathing panel with the ASCE-7 wind loading pattern to determine the wind pressure load on the individual panel.
- **PPpzonecoords.m**  
Defines the coordinates of the pressure zones based on the ASCE-7 wind loading pattern.
- **ILoadRWC.m**  
Calculate a pressure value for each roof to wall connection from the pressure on roof sheathing.
- **PPrwc\_loads.m**  
Given the pressures already calculated on the roof sheathing panels, the roof shape, and the location of the roof to wall connections, determine the loads transferred to the roof to wall connections.
- **ILoadW.m**  
Calculate the load on the walls from the ASCE-7 wind loading equations.
- **ILoadO.m**  
Calculate the load on the windows, doors and garage door using the ASCE-7 wind loading equations.
- **InitialFCheck.m**  
Check for failure of the structural elements Roof Sheathing, Walls, Openings, and Roof to Wall Connections.
- **IFCRS.m**  
Check if Roof Sheathing load is larger than Roof Sheathing strength. If load > strength, mark as failed.
- **IFCW.m**  
Check if wall load is larger than wall strength, if so, mark as failed.
- **IFCO.m**  
Check if load on openings is larger than strength of openings, if so, mark as failed.
- **ImpactSim.m**  
Simulate projectiles flying through the air and the probability of impact with openings. Calculate the number of failed openings from impact.
- **IFCRWCCheck.m**  
Check if load on roof to wall connections load is larger than strength, if so, mark as failed.
- **FinalLoads.m**  
Based on initial failures, recalculate the load on structural elements.
- **Pressure\_Recalc.m**  
If the building envelope has been compromised, recalculate the pressure on the structural elements. If the building envelope is compromised, recalculate the

load on the structural elements, otherwise don't change the load.

→**FLoadO.m**  
Recalculate the load on the openings based on the new pressure.

→**FLoadRS.m**  
Recalculate the load on the roof sheathing elements based on the new pressure  
→**preprocessRS.m**  
→**PPpzonecoords.m**

→**FFCRS.m**  
If load changes, check for roof sheathing failure to inform the roof to wall connection load recalculation.

→**FLoadRC.m**  
Recalculate the load on the roof cover based on the new pressure.  
→**preprocessRS.m**  
→**PPpzonecoords.m**

→**FLoadRWC.m**  
Recalculate the roof to wall connection loads based on the new pressure, new loads on roof sheathing, and failure of roof sheathing.  
→**PPrwc\_loads.m**

→**FFCRWC.m**  
If the load changes, check for roof to wall connection failure to inform the wall load.

→**FLoadW.m**  
Recalculate the load on the walls based on the new pressure and recalculated loads and failure of the roof to wall connections.

→**FinalFCheck2.m**  
Given changes in loading, recheck for failure on the structural elements.  
→**FFCO.m**  
Check if load on openings is larger than strength of openings, if so, mark as failed.

→**FFCRS.m**  
Check if load on roof sheathing elements is larger than strength of roof sheathing elements, if so, mark as failed.

→**FFCRC.m**  
Check if load on roof cover elements is larger than strength of roof cover elements, if so, mark as failed.

→**IFCRWCCheck2.m**  
Check if load on roof to wall connections is larger than strength of roof to wall connections, if so, mark as failed.

→**FLoadWAdjust.m**  
Given roof to wall connection failures, recalculate load on the walls.

→**FFCW.m**  
Check if load on walls is larger than strength of walls, if so, mark as failed.

→**ComponentDamageState.m**

Check the failure of each of the structural components and assign the components a damage state based on the number of failed elements.

→**BuildingDamageState.m**

Combining the component damage states, determine the building damage states.

→**ProbDamageStates.m**

Calculate the probability of the component and building damage states by averaging the number of buildings that enter each damage state for each building type, resistance combination, and wind speed based on the Maxiter simulations of the building for each orientation.

## APPENDIX J

### RESOURCE ALLOCATION HEURISTIC MATLAB CODE

#### ***Run\_RA.m***

```
function []=Run_RA()

foldernameA = '.../AFiles503';
foldernameP = '.../ProcessingMaster';

%%% 1M Run %%%
BudgetPre = 1000000;
BudgetPost = 1000000;
foldername = '.../1MResults';
foldernamePP = '.../1MProcessing';
GA_RA_Fraction_All_2B_lnx(BudgetPre,BudgetPost,foldername,foldernameA,foldernameP,foldernamePP);
```

#### ***GA\_RA\_Fraction\_All\_2B\_lnx.m***

```
function
[]=GA_RA_Fraction_All_2B_lnx(BudgetPre,BudgetPost,foldername,foldernameA,foldernameP,foldernamePP)

global bPre;
global bPost;

%Greedy Heuristic to determine Resource Allocation
%Three Stages: 1) Pre-Disaster Mitigation
%               2) Post-Disaster Mitigation
%               3) Post-Disaster Acquisition

%Inputs:

%i = census tracts
%m = building types
%d = damage states
%t = time periods
%c = component resistance combination
%D(c) = possible mitigations for each component resistance combination c

%p(l) = Probability of occurrence of hurricane event l

%A(i,m,c,d,l) = Proportion of buildings of type m that are designed to
%component resistance combination c that will be damaged to damage state d
%by hurricane l winds that occur in census tract i.

%R(m,c,d,c) = Cost per building to reconstruct buildings of type m that
%are originally designed to component resistance combination c, are damaged
%to damage state d and reconstructed to the same component resistance
%combination c.
```

```

%F(m,c,c') = Cost per building to mitigate buildings of type m from
%component resistance combination c to component resistance combination c'.

%M(m,c,d,c') = Cost per building beyond the reconstruction cost, R, to
%mitigate from component resistance combination c to component resistance
%combination c' when reconstructing from damage state d.

%V(m,c,d) = Cost per building to "buy out" or "acquire" buildings (remove
%the buildings from the building inventory) of type m that are designed to
%component resistance combination c that are damaged to damage state d.

%B() = Budget for mitigation in each time period

%Outputs:

%Z(i,m,c,c',t) = Number of buildings in census tract i of building type m
%that are mitigated from component resistance combination c to component
%resistance combination c' during time period t.

%HM(i,m,c,d,c',t) = Number of buildings in census tract i of building type m
%that are mitigated from component resistance combination c to component
%resistance combination c' after being damaged to damage state d during
%time period t.

%HR(i,m,c,d,c,t) = Number of buildings in census tract i of building type m
%that are rebuilt to component resistance combination c after being damaged
%to damage state d during time period t.

%IA(i,m,c,d,t) = Number of buildings in census tract i of building type m
%that are of component resistance combination c that are removed from the
%building inventory after having been damaged to damage state d in time
%period t.

%X(i,m,c,t) = Number of buildings in census tract i at the end of time
%period t that are of building type m and are designed to component
%resistance combination c.

%Other Variables:
%w(i,m,c) = Intermediate variable to track the Number of buildings in
%census tract i that are of building type m and are designed to component
%resistance combination c.

%Load Inputs
display('Load Inputs');

load('XInitial503.mat'); %Load XInitial

load('CensusTracts503.mat');
global ct; % Number of Census Tracts
ct = length(CensusTracts);
load('BuildingTypes.mat');
global blg; % Number of Building Types
blg = length(BuildingTypes);
load('crc_wf.mat');
global comp; % Number of Component Resistance Combinations
comp = length(CRC);
load('TimeStep.mat');
global time; % Number of Time Steps
time = length(TimeStep);
load('DamageStates.mat');
global dmg;
dmg = length(DamageStates);

```



```

global F;
F = zeros(blg,comp,comp);
FValues = textread('f.csv');
for k = 1:length(FValues)
    F(FValues(k,1),FValues(k,2),FValues(k,3)) = FValues(k,4);
end
global R;
R = zeros(blg,comp,dmg,comp);
RValues = csvread('r.csv');
for k = 1:length(RValues)
    R(RValues(k,1),RValues(k,2),RValues(k,3),RValues(k,4)) = RValues(k,5);
end
global V;
V = zeros(blg,comp,dmg);
VValues = csvread('v.csv');
for m = 1:blg
    for c = 1:comp
        for d = 1:dmg
            V(m,c,d) = VValues(m,2);
        end
    end
end
global M;
M = zeros(blg,comp,dmg,comp);
MValues = csvread('m.csv');
for k = 1:length(MValues)
    M(MValues(k,1),MValues(k,2),MValues(k,3),MValues(k,4)) = MValues(k,5);
end
global p;
load('p_Hur_WF.mat','p');
global hur;
hur = length(p);

global D;
load d;
global c_o;
M_able = sum(D);
c_o = find(M_able);

%Step 1: Initialize Solution
display('Step 1: Initialize Solution');

X = zeros(ct,blg,comp,time+1); %SIZE OK!!!
X(:, :, :, 1) = XInitial; %SIZE OK!!!
BPre = [BudgetPre*ones(time/2,1);zeros(time/2,1)];
BPost = [BudgetPost*ones(time/2,1);zeros(time/2,1)];
unused_bPre = zeros(time,1);
unused_bPost = zeros(time,1);

global w;
w = zeros(ct,blg,comp);

%Reconstruction Costs - Expected value of reconstruction costs for a
%building in census tract i of building type m designed to component
%resistance combination c_o.
global ReconCost;
load(strcat(foldername,'/ReconCost.mat'),'ReconCost');

global Y;
load(strcat(foldername,'/Y.mat'),'Y');

```

```

% Y = zeros(ct,blg,comp,dmg); %SIZE OK!!!
% ReconCost = zeros(ct,blg,comp);
% display(strcat('Y and ReconCost Caluculation'));
ExpLossT = zeros(time,ct,blg,comp);
ExpLossNumT = zeros(time,ct,blg,comp);
% for i0 = 1:ct
%     filename = strcat(foldernameA, '/A_i', num2str(i0), '.mat');
%     load(filename, 'A');
%     display(strcat('ct=', num2str(i0)));
%     for m = 1:blg
%         for orig = 1:length(c_o)
%             for d = 1:dmg
%
%                 Y(i0,m,c_o(orig),d)=
%                     reshape(A(m,c_o(orig),d,:),1,length(p))*p(:);
%
%                 ReconCost(i0,m,c_o(orig)) = ReconCost(i0,m,c_o(orig)) + ...
%                     R(m,c_o(orig),d,c_o(orig))*Y(i0,m,c_o(orig),d);
%
%             end
%         end
%     end
% end
% save(strcat(foldername, '/Y.mat'), 'Y');
% save(strcat(foldername, '/ReconCost.mat'), 'ReconCost');

display(strcat('Exp Loss Calculation'));
for i0 = 1:ct
    display(strcat('ct=', num2str(i0)));
    for m = 1:blg
        for orig = 1:length(c_o)
            for d = 1:dmg
                ExpLossT(:,i0,m,c_o(orig)) = ExpLossT(:,i0,m,c_o(orig)) +
                    R(m,c_o(orig),d,c_o(orig))*Y(i0,m,c_o(orig),d)*XInitial(i
                    0,m,c_o(orig));
                ExpLossNumT(:,i0,m,c_o(orig))=ExpLossNumT(:,i0,m,c_o(orig))+
                    Y(i0,m,c_o(orig),d)*XInitial(i0,m,c_o(orig));
            end
        end
    end
end
end

ExpLossT = ExpLossT(1:time/2,:,:,:);
filename = strcat(foldername, '/ExpLossT.mat');
save(filename, 'ExpLossT');
ExpLossTime = sum(sum(sum(ExpLossT,2),3),4);
ExpLossCT = sum(sum(sum(ExpLossT,1),3),4)';
filename = strcat(foldername, '/ExpLossTime.csv');
csvwrite(filename, ExpLossTime);
filename = strcat(foldername, '/ExpLossCT.csv');
csvwrite(filename, ExpLossCT);

ExpLossNumT = ExpLossNumT(1:time/2,:,:,:);
filename = strcat(foldername, '/ExpLossNumT.mat');
save(filename, 'ExpLossNumT');
ExpLossNumTime = sum(sum(sum(ExpLossNumT,2),3),4);
ExpLossNumCT = sum(sum(sum(ExpLossNumT,1),3),4)';
filename = strcat(foldername, '/ExpLossNumTime.csv');
csvwrite(filename, ExpLossNumTime);
filename = strcat(foldername, '/ExpLossNumCT.csv');
csvwrite(filename, ExpLossNumCT);
%

```

```

% %Step 2: Select Mitigation Alternatives
% display('Step 2: Select Mitigation Alternatives');
% display('ROI Pre Disaster Mitigation');
% %Calculate Return on Investment Pre-Disaster Mitigation Alternatives
% ROIPre = tic;
% RE_Pre_Mit_Frac_All(foldernameP);
% tROIPre = toc(ROIPre);
% display(strcat('Return on Investment PRE MIT Calc
Time:',num2str(tROIPre)));
% %Calculate Return on Investment Post-Disaster Mitigation Alternatives
% display('ROI Post Disaster Mitigation');
% ROIPost = tic;
% RE_Post_Mit_Frac_All(foldernameP);
% tROIPost = toc(ROIPost);
% display(strcat('Return on Investment POST MIT Calc
Time:',num2str(tROIPost)));
% %Calculate Return on Investment Post-Disaster Acquisition Alternatives
% display('ROI Post Disaster Acquisition');
% ROIAcq = tic;
% RE_Post_Acq_Frac_All(foldernameP);
% tROIAcq = toc(ROIAcq);
% display(strcat('Return on Investment ACQ Calc Time:',num2str(tROIAcq)));

%For each time step, determine the best investments of the budget, update
%the inventory at the end of each step, move on to the next time step.
display('Choose Alternatives');
MitCostT = zeros(time,ct,blg,comp);
MitNumT = zeros(time,ct,blg,comp);
MitAfterCostT = zeros(time,ct,blg,comp);
ReconMitCostT = zeros(time,ct,blg,comp);
MitAfterNumT = zeros(time,ct,blg,comp);
AcqCostT = zeros(time,ct,blg,comp);
AcqNumT = zeros(time,ct,blg,comp);
ReconOtherCostT = zeros(time,ct,blg,comp);
ReconOtherNumT = zeros(time,ct,blg,comp);
ExpLossMitT = zeros(time,ct,blg,comp);
ExpLossMitNumT = zeros(time,ct,blg,comp);

% global count;
for t = 1:time/2
    display(t);
    %Initialize intermediate budget value for budget at time t
    bPre = BPre(t);
    bPost = BPost(t);
    %Initialize intermediate inventory as the inventory at the end of
    %previous time step (X(t) = inventory at the end of t-1)
    w(:, :, :) = X(:, :, :, t);

    %Sort the ROI values for Pre, Post and Acquisition categories.
    display('Sort ROI Pre Event Options');
    [ROIPre] = OrderRE_All2BPre(t,foldernameP,foldernamePP);
    display('Done sorting ROI Pre Event Options');

    while sum(ROIPre(:,7))>0 && bPre > 0

        %If all buildings in the inventory have been mitigated/acquired in
        %this time period, update variables and move on to next time step.
        [btm] = find(w);
        if isempty(btm)
            unused_bPre(t) = bPre;
            break;
        end
    end
end

```

```

use_BPre = 0;
%Determine the alternative from the best of the categories
[MaxRE,ROI_Index]=max(ROIPre(:,7));

%If the Pre Damage Mitigation option is chosen
if ROIPre(ROI_Index,1) == 1
    i1=ROIPre(ROI_Index,2); m1=ROIPre(ROI_Index,3);
    col=ROIPre(ROI_Index,4); cfl=ROIPre(ROI_Index,6);

    if w(i1,m1,col) > 0
        [use_BPre]=PreDMit2B(t,i1,m1,col,cfl,foldername);
    end

    %Remove that option from list of options
    ROIPre(ROI_Index,7) = 0;
    if w(i1,m1,col) == 0
        ROIremove = ROIPre(:,2)==i1 &
            ROIPre(:,3)==m1 & ROIPre(:,4)==col;
        ROIPre(ROIremove,7) = 0;

        display(strcat(['Removed ',num2str(sum(ROIremove)), '
            options']));
    end
end

bPre = bPre - use_BPre;

end

display('Calculate HR for each CT');
for i0 = 1:ct
    HR = zeros(blg,comp,dmg);
    display(i0);
    for m = 1:blg
        for c = 1:comp
            for d = 1:dmg
                HR(m,c,d) = Y(i0,m,c,d)*w(i0,m,c);
            end
        end
    end
    display(strcat('Saving HR for CT ',num2str(i0)));
    save(strcat(foldername,'/HR_t_',num2str(t),'_i_',num2str(i0),'.mat'),
        'HR');
end

display('Sort ROI Post Event Options');
[ROIPost] = OrderRE_All2BPost(t,foldernameP,foldernamePP,foldername);
display('Done sorting ROI Post Event Options');

%Evaluate investmets while Budget is positive and viable options
%(ROI>0) exist
while sum(ROIPost(:,7))>0 && bPost > 0

    %If all buildings in the inventory have been mitigated/acquired in
    %this time period, update variables and move on to next time step.
    [btm] = find(w);
    if isempty(btm)
        unused_bPost(t) = bPost;
        break;
    end
    use_BPost = 0;
    %Determine the alternative from the best of the categories

```

```

[MaxRE,ROI_Index]=max(ROIPost(:,7));

%If the Post Damage Mitigation option is chosen
if ROIPost(ROI_Index,1) == 2
    i1=ROIPost(ROI_Index,2); m1=ROIPost(ROI_Index,3);
    col=ROIPost(ROI_Index,4);
    d1=ROIPost(ROI_Index,5); cfl=ROIPost(ROI_Index,6);

    %If there are buildings that might be damaged in this way in
    %this time step
    load(strcat(foldername,'/HR_t_',num2str(t),'_i_',num2str(i1),'.m
        at'),'HR');
    if HR(m1,col,d1) > 0
        [use_BPost]=PostDMit2B(t,i1,m1,col,d1,cfl,foldername);
    else
        ROIremove = ROIPost(:,2)==i1 & ROIPost(:,3)==m1 & ...
            ROIPost(:,4)==col & ROIPost(:,5)==d & ROIPost(:,7)>0;
        if sum(ROIremove) > 0
            ROIPost(ROIremove,7) = 0;
            ROIleft = ROIPost(:,7)>0;
            display(strcat(['Removed ',num2str(sum(ROIremove)),'
                options, ',num2str(sum(ROIleft)),' options left']));
        end
    end
end

%Remove that option from list of options
ROIPost(ROI_Index,7) = 0;
if w(i1,m1,col) == 0
    ROIremove = ROIPost(:,2)==i1 & ROIPost(:,3)==m1 &
        ROIPost(:,4)==col;
    if sum(ROIremove) > 0
        ROIPost(ROIremove,7) = 0;
        ROIleft = ROIPost(:,7)>0;
        display(strcat(['Removed ',num2str(sum(ROIremove)),'
            options, ',num2str(sum(ROIleft)),' options left']));
    end
end

%If the Post Damage Acquisition option is chosen
elseif ROIPost(ROI_Index,1) == 3
    i1=ROIPost(ROI_Index,2); m1=ROIPost(ROI_Index,3);
    col=ROIPost(ROI_Index,4);
    d1=ROIPost(ROI_Index,5);

    %If there are buildings that might be damaged in this way in
    %this time step
    load(strcat(foldername,'/HR_t_',num2str(t),'_i_',num2str(i1),'.ma
        t'),'HR');
    if HR(m1,col,d1) > 0
        [use_BPost]=PostDAcq2B(t,i1,m1,col,d1,foldername);
    else
        ROIremove = ROIPost(:,2)==i1 & ROIPost(:,3)==m1 & ...
            ROIPost(:,4)==col & ROIPost(:,5)==d & ROIPost(:,7)>0;
        if sum(ROIremove) > 0
            ROIPost(ROIremove,7) = 0;
            ROIleft = ROIPost(:,7)>0;
            display(strcat(['Removed ',num2str(sum(ROIremove)),'
                options, ',num2str(sum(ROIleft)),' options left']));
        end
    end
end

```

```

%Remove that option from list of options

ROIPost(ROI_Index,7) = 0;
if w(i1,m1,col) == 0
    ROIremove = ROIPost(:,2)==i1 & ROIPost(:,3)==m1 &
        ROIPost(:,4)==col;
    if sum(ROIremove)>0
        ROIPost(ROIremove,7) = 0;
        ROIleft = ROIPost(:,7)>0;
        display(strcat(['Removed ',num2str(sum(ROIremove)),'
            options, ',num2str(sum(ROIleft)),' options left']));
    end
end
end

%Update the budget
bPost = bPost - use_BPost;

end

display('Updating Inventory');
%Update inventory for the start of time t+1.
%New Inventory X(i0,m,c,t+1) = Inventory at start of t X(i0,m,c,t) -
%Inventory mitigated pre damage FROM c to c' Z(i0,m,c,c',t) +
%Inventory mitigated pre damage TO c from c'' Z(i0,m,c'',c,t) -
%Inventory mitigated post damage FROM c to c' H(i0,m,c,c',t) +
%Inventory mitigated post damage TO c from c'' H(i0,m,c'',c,t) -
%Inventory removed via acquisition post damage IA(i0,m,c,d,t)

PreMitTo = zeros(ct,blg,comp);
PreMitFrom = zeros(ct,blg,comp);
display('Sum up Mitigated Buildings, and Calculate Mitigation Cost');
for i0 = 1:ct
    filename =
        strcat(foldername,'/Z_t_',num2str(t),'_i_',num2str(i0),'.mat');
    if exist(filename,'file') == 2
        load(filename,'Z');
        for m = 1:blg
            for orig = 1:length(c_o)
                %Inventory mitigated pre damage TO c
                PreMitTo(i0,m,c_o(orig)) = sum(Z(m, :, c_o(orig)));
                MitCostT(t,i0,m,c_o(orig)) =
                    sum(F(m,c_o(orig),:).*Z(m,c_o(orig),:));
                MitNumT(t,i0,m,c_o(orig)) = sum(Z(m,c_o(orig),:));
                %Inventory mitigated pre damage FROM c
                PreMitFrom(i0,m,c_o(orig)) = sum(Z(m,c_o(orig),:));
            end
        end
    end
end
end

display('Sum up Post-Mitigated Buildings, and Calculate Post-Mitigation
Cost');
PostMitTo = zeros(ct,blg,comp);
PostMitFrom = zeros(ct,blg,comp);

for i0 = 1:ct
    display(strcat('ct=',num2str(i0)));
    filename =
        strcat(foldername,'/HM_t_',num2str(t),'_i_',num2str(i0),'.mat');
    if exist(filename,'file') == 2
        load(filename,'HM');
    end
end

```

```

        for m = 1:blg
            for orig = 1:length(c_o)
                c_f = find(D(:,c_o(orig)));
                for final = 1:length(c_f)
                    %Inventory mitigated post damage TO c
                    PostMitTo(i0,m,c_f(final)) =
                        PostMitTo(i0,m,c_f(final)) +
                        sum(HM(m,c_o(orig),:,c_f(final)));
                    MitAfterCostT(t,i0,m,c_o(orig)) =
                        MitAfterCostT(t,i0,m,c_o(orig)) +
                        sum(M(m,c_o(orig),:,c_f(final)).*HM(m,c_o(orig),
                            :,c_f(final)));
                    ReconMitCostT(t,i0,m,c_o(orig)) =
                        ReconMitCostT(t,i0,m,c_o(orig)) +
                        sum(R(m,c_o(orig),:,c_f(final)).*HM(m,c_o(orig),
                            :,c_f(final)));
                    MitAfterNumT(t,i0,m,c_o(orig)) =
                        MitAfterNumT(t,i0,m,c_o(orig)) + ...
                        sum(HM(m,c_o(orig),:,c_f(final)));
                end
                c_f2 = find(D(:,c_o(orig)));
                for final2 = 1:length(c_f2)
                    %Inventory mitigated post damage FROM c
                    PostMitFrom(i0,m,c_o(orig)) =
                        PostMitFrom(i0,m,c_o(orig)) +
                        sum(HM(m,c_o(orig),:,c_f2(final2)));
                end
            end
        end
    end
end

display('Sum up Acquired Buildings, and Calculate Acquisition Cost');
AcquiredFrom = zeros(ct,blg,comp);
for i0 = 1:ct
    display(strcat('ct=',num2str(i0)));
    filename =
        strcat(foldername,'/IA_t_',num2str(t),'_i_',num2str(i0),'.mat');
    if exist(filename,'file') == 2
        load(filename,'IA');
        for m = 1:blg
            for orig = 1:length(c_o)
                %Inventory removed via acquisition post damage
                AcquiredFrom(i0,m,c_o(orig)) = sum(IA(m,c_o(orig),:));
                AcqCostT(t,i0,m,c_o(orig)) =
                    sum(V(m,c_o(orig),:).*IA(m,c_o(orig),:));
                AcqNumT(t,i0,m,c_o(orig)) = sum(IA(m,c_o(orig),:));
            end
        end
    end
end

X(:, :, :, t+1) = X(:, :, :, t) -...
    PreMitFrom(:, :, :)+PreMitTo(:, :, :)-...
    PostMitFrom(:, :, :)+PostMitTo(:, :, :)-...
    AcquiredFrom(:, :, :);

for i0 = 1:ct
    filename =
        strcat(foldername,'/HR_t_',num2str(t),'_i_',num2str(i0),'.mat');
    if exist(filename,'file') == 2
        load(filename,'HR');
    end
end

```

```

        for m = 1:blg
            for final = 1:length(c_o)
                for d = 1:dmg
                    ReconOtherCostT(t,i0,m,c_o(final)) =
                        ReconOtherCostT(t,i0,m,c_o(final)) +
                        R(m,c_o(final),d,c_o(final))*HR(m,c_o(final),d);
                    ReconOtherNumT(t,i0,m,c_o(final)) =
                        ReconOtherNumT(t,i0,m,c_o(final)) +
                        HR(m,c_o(final),d);
                end
            end
        end
    end
end

for i0 = 1:ct
    for m = 1:blg
        for final = 1:length(c_o)
            ExpLossMitT(t,i0,m,c_o(final)) =
                ReconOtherCostT(t,i0,m,c_o(final)) +
                ReconMitCostT(t,i0,m,c_o(final));
            ExpLossMitNumT(t,i0,m,c_o(final)) =
                ReconOtherNumT(t,i0,m,c_o(final)) +
                MitAfterNumT(t,i0,m,c_o(final));
        end
    end
end

filename = strcat(foldername, '/MitCostT.mat');
save(filename, 'MitCostT');
filename = strcat(foldername, '/MitNumT.mat');
save(filename, 'MitNumT');
filename = strcat(foldername, '/MitAfterCostT.mat');
save(filename, 'MitAfterCostT');
filename = strcat(foldername, '/MitAfterNumT.mat');
save(filename, 'MitAfterNumT');
filename = strcat(foldername, '/ReconMitCostT.mat');
save(filename, 'ReconMitCostT');
filename = strcat(foldername, '/AcqCostT.mat');
save(filename, 'AcqCostT');
filename = strcat(foldername, '/AcqNumT.mat');
save(filename, 'AcqNumT');
filename = strcat(foldername, '/ReconOtherCostT.mat');
save(filename, 'ReconOtherCostT');
filename = strcat(foldername, '/ReconOtherNumT.mat');
save(filename, 'ReconOtherNumT');
filename = strcat(foldername, '/ExpLossMitT.mat');
save(filename, 'ExpLossMitT');
filename = strcat(foldername, '/ExpLossMitNumT.mat');
save(filename, 'ExpLossMitNumT');
filename = strcat(foldername, '/X.mat');
save(filename, 'X');

end

MitCostT = MitCostT(1:time/2, :, :, :);
filename = strcat(foldername, '/MitCostT.mat');
save(filename, 'MitCostT');
MitCostTime = sum(sum(sum(MitCostT, 2), 3), 4);
MitCostCT = sum(sum(sum(MitCostT, 1), 3), 4)';
filename = strcat(foldername, '/MitCostTime.csv');
csvwrite(filename, MitCostTime);

```



```

filename = strcat(foldername, '/MitCostCT.csv');
csvwrite(filename, MitCostCT);

MitNumT = MitNumT(1:time/2, :, :, :);
filename = strcat(foldername, '/MitNumT.mat');
save(filename, 'MitNumT');
MitNumTime = sum(sum(sum(MitNumT, 2), 3), 4);
MitNumCT = sum(sum(sum(MitNumT, 1), 3), 4)';
filename = strcat(foldername, '/MitNumTime.csv');
csvwrite(filename, MitNumTime);
filename = strcat(foldername, '/MitNumCT.csv');
csvwrite(filename, MitNumCT);

MitAfterCostT = MitAfterCostT(1:time/2, :, :, :);
filename = strcat(foldername, '/MitAfterCostT.mat');
save(filename, 'MitAfterCostT');
MitAfterCostTime = sum(sum(sum(MitAfterCostT, 2), 3), 4);
MitAfterCostCT = sum(sum(sum(MitAfterCostT, 1), 3), 4)';
filename = strcat(foldername, '/MitAfterCostTime.csv');
csvwrite(filename, MitAfterCostTime);
filename = strcat(foldername, '/MitAfterCostCT.csv');
csvwrite(filename, MitAfterCostCT);

MitAfterNumT = MitAfterNumT(1:time/2, :, :, :);
filename = strcat(foldername, '/MitAfterNumT.mat');
save(filename, 'MitAfterNumT');
MitAfterNumTime = sum(sum(sum(MitAfterNumT, 2), 3), 4);
MitAfterNumCT = sum(sum(sum(MitAfterNumT, 1), 3), 4)';
filename = strcat(foldername, '/MitAfterNumTime.csv');
csvwrite(filename, MitAfterNumTime);
filename = strcat(foldername, '/MitAfterNumCT.csv');
csvwrite(filename, MitAfterNumCT);

ReconMitCostT = ReconMitCostT(1:time/2, :, :, :);
filename = strcat(foldername, '/ReconMitCostT.mat');
save(filename, 'ReconMitCostT');
ReconMitCostTime = sum(sum(sum(ReconMitCostT, 2), 3), 4);
ReconMitCostCT = sum(sum(sum(ReconMitCostT, 1), 3), 4)';
filename = strcat(foldername, '/ReconMitCostTime.csv');
csvwrite(filename, ReconMitCostTime);
filename = strcat(foldername, '/ReconMitCostCT.csv');
csvwrite(filename, ReconMitCostCT);

AcqCostT = AcqCostT(1:time/2, :, :, :);
filename = strcat(foldername, '/AcqCostT.mat');
save(filename, 'AcqCostT');
AcqCostTime = sum(sum(sum(AcqCostT, 2), 3), 4);
AcqCostCT = sum(sum(sum(AcqCostT, 1), 3), 4)';
filename = strcat(foldername, '/AcqCostTime.csv');
csvwrite(filename, AcqCostTime);
filename = strcat(foldername, '/AcqCostCT.csv');
csvwrite(filename, AcqCostCT);

AcqNumT = AcqNumT(1:time/2, :, :, :);
filename = strcat(foldername, '/AcqNumT.mat');
save(filename, 'AcqNumT');
AcqNumTime = sum(sum(sum(AcqNumT, 2), 3), 4);
AcqNumCT = sum(sum(sum(AcqNumT, 1), 3), 4)';
filename = strcat(foldername, '/AcqNumTime.csv');
csvwrite(filename, AcqNumTime);
filename = strcat(foldername, '/AcqNumCT.csv');
csvwrite(filename, AcqNumCT);

```

```

ReconOtherCostT = ReconOtherCostT(1:time/2, :, :, :);
filename = strcat(foldername, '/ReconOtherCostT.mat');
save(filename, 'ReconOtherCostT');
ReconOtherCostTime = sum(sum(sum(ReconOtherCostT, 2), 3), 4);
ReconOtherCostCT = sum(sum(sum(ReconOtherCostT, 1), 3), 4)';
filename = strcat(foldername, '/ReconOtherCostTime.csv');
csvwrite(filename, ReconOtherCostTime);
filename = strcat(foldername, '/ReconOtherCostCT.csv');
csvwrite(filename, ReconOtherCostCT);

ReconOtherNumT = ReconOtherNumT(1:time/2, :, :, :);
filename = strcat(foldername, '/ReconOtherNumT.mat');
save(filename, 'ReconOtherNumT');
ReconOtherNumTime = sum(sum(sum(ReconOtherNumT, 2), 3), 4);
ReconOtherNumCT = sum(sum(sum(ReconOtherNumT, 1), 3), 4)';
filename = strcat(foldername, '/ReconOtherNumTime.csv');
csvwrite(filename, ReconOtherNumTime);
filename = strcat(foldername, '/ReconOtherNumCT.csv');
csvwrite(filename, ReconOtherNumCT);

ExpLossMitT = ExpLossMitT(1:time/2, :, :, :);
filename = strcat(foldername, '/ExpLossMitT.mat');
save(filename, 'ExpLossMitT');
ExpLossMitTime = sum(sum(sum(ExpLossMitT, 2), 3), 4);
ExpLossMitCT = sum(sum(sum(ExpLossMitT, 1), 3), 4)';
filename = strcat(foldername, '/ExpLossMitTime.csv');
csvwrite(filename, ExpLossMitTime);
filename = strcat(foldername, '/ExpLossMitCT.csv');
csvwrite(filename, ExpLossMitCT);

ExpLossMitNumT = ExpLossMitNumT(1:time/2, :, :, :);
filename = strcat(foldername, '/ExpLossMitNumT.mat');
save(filename, 'ExpLossMitNumT');
ExpLossMitNumTime = sum(sum(sum(ExpLossMitNumT, 2), 3), 4);
ExpLossMitNumCT = sum(sum(sum(ExpLossMitNumT, 1), 3), 4)';
filename = strcat(foldername, '/ExpLossMitNumTime.csv');
csvwrite(filename, ExpLossMitNumTime);
filename = strcat(foldername, '/ExpLossMitNumCT.csv');
csvwrite(filename, ExpLossMitNumCT);

display('Step 3: Return Decision Variables X,Z,H,I');
X = X(:, :, :, 1:time/2+1);
filename = strcat(foldername, '/X.mat');
save(filename, 'X');

```

### ***RE\_Pre\_Mit\_Frac\_All.m***

```

function [tPMitTic, tLMitTic, tRMitTic] = RE_Pre_Mit_Frac_All(foldernameP)
%Calculates the Return on Investment (reduction in estimated future losses)
%of the mitigation. ROI = Difference between costs without and with
%mitigation, divided by the Cost of the mitigation.

global ct; global blg; global comp; global c_o; global D; global time;
global F;

%P(i,m,c,c') = cost associated with mitigation of buildings of type m in
%census tract i from resistance combination c to combination c'
display('P Value for Mitigation');

```

```

PMitTic = tic;
P = zeros(ct,blg,comp,comp);
for i0 = 1:ct
    for m = 1:blg
        for orig = 1:length(c_o)
            c_f = find(D(:,c_o(orig)));
            for final = 1:length(c_f)
                P(i0,m,c_o(orig),c_f(final)) = ...
                    P_Pre_Mit_Frac(i0,m,c_o(orig),c_f(final)); %SIZE OK
            end
        end
    end
end
end
tPMitTic = toc(PMitTic);
display(strcat('Time to Calc P Values for PRE MIT:',num2str(tPMitTic)));

%L(i,m,c,c',t) = cost associated with mitigating a building from c to c' in
%time period t, and then reconstructing damage to that building in that
%time period and all future time periods in the time horizon.
display('L Value for Mitigation');
LMitTic = tic;

for t = 1:time/2
    display(strcat('PreMit L TimeStep = ',num2str(t)));
    L = zeros(ct,blg,comp,comp);
    for i0 = 1:ct
        for m = 1:blg
            for orig = 1:length(c_o)
                c_f = find(D(:,c_o(orig)));
                for final = 1:length(c_f)
                    L(i0,m,c_o(orig),c_f(final)) = ...
                        P(i0,m,c_o(orig),c_f(final)) + ...
                        (time - t)*P(i0,m,c_f(final),c_f(final));
                end
            end
        end
    end
    end
    filename = strcat(foldernameP,'/L_Pre_t_',num2str(t),'.mat');
    save(filename,'L');
end
tLMitTic = toc(LMitTic);
display(strcat('Time to Calc L Values for PRE MIT:',num2str(tLMitTic)));

%RE_Pre(i,m,c,c',t) = Return on Investment of mitigation building of type m
%in census tract i, from resistance combination c to c'. The ROI is the
%difference between the L value with and without mitigation in time t,
%divided by the cost of the mitigation.
display('ROI Value for Mitigation');
RMitTic = tic;

for t = 1:time/2
    display(strcat('PreMit R TimeStep = ',num2str(t)));
    RE_Pre = zeros(ct,blg,comp,comp);
    filenameL = strcat(foldernameP,'/L_Pre_t_',num2str(t),'.mat');
    load(filenameL);
    for i0 = 1:ct
        for m = 1:blg
            for orig = 1:length(c_o)
                c_f = find(D(:,c_o(orig)));
                for final = 1:length(c_f)
                    RE_Pre(i0,m,c_o(orig),c_f(final)) = ...

```

```

                                (L(i0,m,c_o(orig),c_o(orig)) - ...
                                L(i0,m,c_o(orig),c_f(final))) / ...
                                F(m,c_o(orig),c_f(final));
                                end
                                end
                                end
                                end
                                filename = strcat(foldernameP,'/RE_Pre_t_',num2str(t),'.mat');
                                save(filename,'RE_Pre');
                                end
                                tRMitTic = toc(RMitTic);
                                display(strcat('Time to Calc ROI Values for PRE MIT:',num2str(tRMitTic)));

```

### ***P\_Pre\_Mit\_Frac.m***

```

function [P]=P_Pre_Mit_Frac(i,m,orig,final)

global F; global ReconCost;

%Mitigation Cost
if orig == final
    Mit_Cost = 0;
else
    Mit_Cost = F(m,orig,final);
end

%Reconstruction Cost
P = Mit_Cost + ReconCost(i,m,orig);

```

### ***RE\_Post\_Mit\_Frac\_All.m***

```

function [Time_LTimeStepPost,tLPostMitTic,...
    Time_RTimeStepPost,tRPostMitTic]=RE_Post_Mit_Frac_All(foldernameP)
%Calculates the Return on Investment (reduction in estimated future losses)
%of mitigating a home when reconstructing it from damage in a given time
%period. ROI = Difference between costs without and with mitigation,
%divided by the Cost of the mitigation.

global ct; global blg; global c_o; global D; global time;
global M; global R; global dmg; global comp; global ReconCost;
% global foldernameP;

%L(i,m,c,d,c',t) = cost associated with mitigating a building from c to c' in
%time period t after the building has been damaged to damage state d, and
%then reconstructing damage to that building in that time period and all
%future time periods in the time horizon.
display('L Value for Mitigation');
LPostMitTic = tic;
Time_LTimeStepPost = zeros(time,1);
for t = 1:time/2
    display(strcat('PostMit L TimeStep = ',num2str(t)));
    LTimeStepPost = tic;
    for i0 = 1:ct
        L = zeros(blg,comp,dmg,comp);
        for m = 1:blg
            for orig = 1:length(c_o)
                for d = 1:dmg

```

```

        c_f = find(D(:,c_o(orig)));
        for final = 1:length(c_f)
            L(m,c_o(orig),d,c_f(final)) = ...
                M(m,c_o(orig),d,c_f(final)) + ...
                R(m,c_o(orig),d,c_f(final)) + ...
                (time-t)*ReconCost(i0,m,c_f(final));
        end
    end
end
filename =
    strcat(foldernameP,'/L_Post_t_',num2str(t),'_i_',num2str(i0),'.'
        'mat');
    save(filename,'L');
end
Time_LTimeStepPost(t) = toc(LTimeStepPost);
display(strcat('Time = ',num2str(Time_LTimeStepPost(t))));
end
tLPostMitTic = toc(LPostMitTic);
display(strcat('Time to Calc L Values for POST MIT:',num2str(tLPostMitTic)));

%RE_Post(i,m,c,d,c',t) = Return on Investment of mitigation building of type
%m in census tract i, from resistance combination c to c' after the building
%has been damaged to damage state d. The ROI is the difference between the
%L value with and without mitigation in time t, divided by the cost of the
%mitigation (beyond the cost of the reconstruction of original damage).
display('ROI Value for Mitigation');
RPostMitTic = tic;

Time_RTimeStepPost = zeros(time,1);
for t = 1:time/2
    display(strcat('PostMit R TimeStep = ',num2str(t)));
    RTimeStepPost = tic;
    for i0 = 1:ct
        RE_Post = zeros(blg,comp,dmg,comp);
        filenameL =
            strcat(foldernameP,'/L_Post_t_',num2str(t),'_i_',num2str(i0),'.'
                'mat');
        load(filenameL);
        for m = 1:blg
            for orig = 1:length(c_o)
                for d = 1:dmg
                    c_f = find(D(:,c_o(orig)));
                    for final = 1:length(c_f)
                        Cost = max(M(m,c_o(orig),d,c_f(final)),1);
                        RE_Post(m,c_o(orig),d,c_f(final)) = ...
                            (L(m,c_o(orig),d,c_o(orig)) - ...
                                L(m,c_o(orig),d,c_f(final)))/...
                                Cost;
                    end
                end
            end
        end
    end
    filename =
        strcat(foldernameP,'/RE_Post_t_',num2str(t),'_i_',num2str(i0),'.'
            'mat');
    save(filename,'RE_Post');
end
Time_RTimeStepPost(t) = toc(RTimeStepPost);
display(strcat('Time = ',num2str(Time_RTimeStepPost(t))));
end
tRPostMitTic = toc(RPostMitTic);

```

```
display(strcat('Time to Calc R Values for POST MIT:',num2str(tRPostMitTic)));
```

### ***RE\_Post\_Acq\_Frac\_All.m***

```
function
[Time_LTimeStepAcq,tLAcqTic,Time_RTimeStepAcq,tRAcqTic]=RE_Post_Acq_Frac_All(
foldernameP)
%Calculates the Return on Investment (reduction in estimated future losses)
%of removing a home from the building inventory by government acquisition
%of the building after it has been damaged in a given time period. ROI =
%Difference between costs of future damage and reconstruction and the cost
%of acquiring the home, divided by the cost of the aquisition.

global ct; global blg; global c_o; global dmg; global time;
global R; global V; global comp; global ReconCost;
% global foldernameP;

%L(i,m,c,d,t) = cost associated with losses in the current and all future
%time periods when a building is neither mitigated or acquired. The cost is
%the reconstruction costs of a building in census tract i of building type
%m, designed to component resistance combination c, and damaged to damage
%state d in time period t and the estimated reconstruction for all future
%time periods in the time horizon.
display('L Value for Acquisition');
LAcqTic = tic;
Time_LTimeStepAcq = zeros(time,1);
for t = 1:time/2
    L = zeros(ct,blg,comp,dmg);
    display(strcat('Acq L TimeStep = ',num2str(t)));
    LTimeStepAcq = tic;
    for i0 = 1:ct
        for m = 1:blg
            for orig = 1:length(c_o)
                for d = 1:dmg
                    L(i0,m,c_o(orig),d) = R(m,c_o(orig),d,c_o(orig)) +...
                        (time-t)*ReconCost(i0,m,c_o(orig));
                end
            end
        end
    end
    filename = strcat(foldernameP,'/L_Acq_t_',num2str(t),'.mat');
    save(filename,'L');
    Time_LTimeStepAcq(t) = toc(LTimeStepAcq);
    display(strcat('Time = ',num2str(Time_LTimeStepAcq(t))));
end
tLAcqTic = toc(LAcqTic);
display(strcat('Time to Calc L Values for ACQ:',num2str(tLAcqTic)));

%RE_Acq(i,m,c,d,c',t) = Return on Investment of removing a building from the
%building inventory that is of building type m in census tract i, from
%resistance combination c after the building has been damaged to damage
%state d. The ROI is the difference between the L value and the cost to
%acquire the building, divided by the cost of the acquisition.
display('ROI Value for Acquisition');
RAcqTic = tic;

Time_RTimeStepAcq = zeros(time,1);
for t = 1:time/2
    RE_Acq = zeros(ct,blg,comp,dmg);
```

```

display(strcat('Acq R TimeStep = ',num2str(t)));
RTimeStepAcq = tic;
filenameL = strcat(foldernameP, '/L_Acq_t_', num2str(t), '.mat');
load(filenameL);
for i0 = 1:ct
    for m = 1:blg
        for orig = 1:length(c_o)
            for d = 1:dmg
                RE_Acq(i0,m,c_o(orig),d) = ...
                    (L(i0,m,c_o(orig),d) - ...
                     V(m,c_o(orig),d))/V(m,c_o(orig),d);
            end
        end
    end
end
filename = strcat(foldernameP, '/RE_Acq_t_', num2str(t), '.mat');
save(filename, 'RE_Acq');
Time_RTimeStepAcq(t) = toc(RTimeStepAcq);
display(strcat('Time = ', num2str(Time_RTimeStepAcq(t))));
end
tRAcqTic = toc(RAcqTic);
display(strcat('Time to Calc R Values for ACQ:', num2str(tRAcqTic)));

```

### ***OrderRE\_All2BPre.m***

```

function [ROIPre] = OrderRE_All2BPre(t, foldernameP, foldernamePP)

display('Order Return on Investments');

display('Pre Disaster Mitigation');
[PreM] = PreMROI(t, foldernameP);

%Combine into one matrix
ROIPre = PreM;

```

### ***PreMROI.m***

```

function [PreM] = PreMROI(t, foldernameP)

global ct; global blg; global comp; global c_o; global D; global w;

PreM = zeros(ct*blg*124,7);
filename = strcat(foldernameP, '/RE_Pre_t_', num2str(t), '.mat');
load(filename, 'RE_Pre');
countPre = 0;

for i0 = 1:ct
    for m = 1:blg
        for orig = 1:length(c_o)
            if w(i0,m,c_o(orig))>0
                c_f = find(D(:,c_o(orig)));
                for final = 1:length(c_f)
                    countPre = countPre + 1;
                    PreM(countPre,:) =
                        [1,i0,m,c_o(orig),0,c_f(final),RE_Pre(i0,m,c_o(orig),
                        c_f(final))];
                end
            end
        end
    end
end

```

```

        end
    end
end
end

%Remove excess rows in matrix
PreM = PreM(1:countPre,:);
PreMRemove = isnan(PreM(:,end)) | PreM(:,end)<0;
PreM(PreMRemove,:) = [];

```

### ***PreDMit2B.m***

```

function [use_BPre]=PreDMit2B(t,i0,m,co,cf,foldername)

global bPre; global blg; global comp; global F; global w;

filename = strcat(foldername,'/Z_t_',num2str(t),'_i_',num2str(i0),'.mat');
if exist(filename,'file') == 0
    Z = zeros(blg,comp,comp);
else
    load(filename,'Z');
end

%Cost to mitigate all buildings in census tract
%If cost exceeds budget, then only mitigate as many possible
%given the budget restriction, update Z variable
Z(m,co,cf) = min(floor(w(i0,m,co)),floor(bPre/F(m,co,cf)));
if Z(m,co,cf) > 0
    use_BPre = Z(m,co,cf)*F(m,co,cf);

    %Update the inventories of co and cf
    w(i0,m,co) = w(i0,m,co)-Z(m,co,cf);
    w(i0,m,cf) = w(i0,m,cf)+Z(m,co,cf);

    save(filename,'Z');
else
    use_BPre = 0;
end

```

### ***OrderRE\_All2BPost.m***

```

function [ROIPost] = OrderRE_All2BPost(t,foldernameP,foldernamePP,foldername)

display('Order Return on Investments');
%Reformat PostMitigation ROI into a two mensional matrix, PostM.
display('Post Disaster Mitigation');
[PostM] = PostMROI(t,foldernameP,foldernamePP,foldername);

display('Acquisition');
[AcqM] = AcqMROI(t,foldernameP,foldername);

%Combine into one matrix
ROIPost = [PostM;AcqM];

```



## ***PostMROI.m***

```
function [PostM] = PostMROI(t,foldernameP,foldernamePP,foldername)

global ct; global blg; global comp; global c_o; global D; global dmg;

PostMTotallength = zeros(ct,1);
for i0 = 1:ct

load(strcat(foldername,'/HR_t_',num2str(t),'_i_',num2str(i0),'.mat'),'HR');
display(strcat('ct = ',num2str(i0)));
countPost = 0;
PostM = zeros(blg*comp*dmg*comp,7);
filename =
strcat(foldernameP,'/RE_Post_t_',num2str(t),'_i_',num2str(i0),'.mat');
load(filename,'RE_Post');
for m = 1:blg
    for orig = 1:length(c_o)
        for d = 1:dmg
            if HR(m,c_o(orig),d)>0
                c_f = find(D(:,c_o(orig)));
                for final = 1:length(c_f)
                    countPost = countPost + 1;
                    PostM(countPost,:) =
                        [2,i0,m,c_o(orig),d,c_f(final),RE_Post(m,c_o(orig),d,c_f(final))];
                end
            end
        end
    end
end

%Remove excess rows in matrix
PostM = PostM(1:countPost,:);
PostMRemove = isnan(PostM(:,end)) | PostM(:,end)<=0;
PostM(PostMRemove,:) = [];
[rows,cols]=size(PostM);
PostMTotallength(i0,1) = rows;
%Save PostM File
filename = strcat(foldernamePP,'/PostM_i_',num2str(i0),'.mat');
save(filename,'PostM');
end

PostMTotal = zeros(sum(PostMTotallength),7);
for i0 = 1:ct
    filename = strcat(foldernamePP,'/PostM_i_',num2str(i0),'.mat');
    load(filename,'PostM');
    indexstart = max(1,(sum(PostMTotallength(1:i0-1))+1));
    indexend = sum(PostMTotallength(1:i0));
    PostMTotal(indexstart:indexend,:) = PostM;
end

PostM = PostMTotal;
```

### ***AcqMROI.m***

```
function [AcqM] = AcqMROI(t,foldernameP,foldername)

global ct; global blg; global comp; global c_o; global dmg;

AcqM = zeros(ct*blg*comp*dmg,7);

filename = strcat(foldernameP,'/RE_Acq_t_',num2str(t),'.mat');
load(filename,'RE_Acq');
Acq = RE_Acq(:,:,,:);
countAcq = 0;

%Reformat PostAcquisition ROI into a two mensional matrix, AcqM.
for i0 = 1:ct

load(strcat(foldername,'/HR_t_',num2str(t),'_i_',num2str(i0),'.mat'),'HR');
    for m = 1:blg
        for orig = 1:length(c_o)
            for d = 1:dmg
                if HR(m,c_o(orig),d)>0
                    countAcq = countAcq + 1;
                    AcqM(countAcq,:) = ...
                        [3,i0,m,c_o(orig),d,0,Acq(i0,m,c_o(orig),d)];
                end
            end
        end
    end
end

AcqM = AcqM(1:countAcq,:);
AcqMRemove = isnan(AcqM(:,end)) | AcqM(:,end)<0;
AcqM(AcqMRemove,:) = [];
```

### ***PostDMit2B.m***

```
function [use_BPost]=PostDMit2B(t,i0,m,co,d,cf,foldername)

global bPost; global M; global blg; global comp; global dmg; global w;
global Y;

filename = strcat(foldername,'/HM_t_',num2str(t),'_i_',num2str(i0),'.mat');
if exist(filename,'file') == 0
    HM = zeros(blg,comp,dmg,comp);
else
    load(filename,'HM');
end
%Cost to mitigate all buildings in census tract
load(strcat(foldername,'/HR_t_',num2str(t),'_i_',num2str(i0),'.mat'),'HR');
%If cost exceeds budget, then only mitigate as many as you can
%given the budget restriction, update H variable
cost = max(1,M(m,co,d,cf));
HM(m,co,d,cf) = min(HR(m,co,d),bPost/cost);
use_BPost = HM(m,co,d,cf)*M(m,co,d,cf);
wupdate = min(HM(m,co,d,cf),w(i0,m,co));

%Update the inventory
w(i0,m,co) = w(i0,m,co)-wupdate;
HR(m,co,d) = HR(m,co,d)-HM(m,co,d,cf);
```

```

save(strcat(foldername, '/HR_t_', num2str(t), '_i_', num2str(i0), '.mat'), 'HR');

save(filename, 'HM');

```

### ***PostDAcq2B.m***

```

function [use_BPost]=PostDAcq2B(t,i0,m,co,d,foldername)

global bPost; global blg; global comp; global dmg; global V; global w;
global Y;

filename = strcat(foldername, '/IA_t_', num2str(t), '_i_', num2str(i0), '.mat');
if exist(filename, 'file') == 0
    IA = zeros(blg, comp, dmg);
else
    load(filename, 'IA');
end
%Cost to mitigate all buildings in census tract
load(strcat(foldername, '/HR_t_', num2str(t), '_i_', num2str(i0), '.mat'), 'HR');
%If cost exceeds budget, then only mitigate as many as you can
%given the budget restriction, update Z variable
IA(m,co,d) = min(HR(m,co,d), bPost/V(m,co,d));
use_BPost = IA(m,co,d)*V(m,co,d);
wupdate = min(IA(m,co,d), w(i0,m,co));

%Update the inventory
w(i0,m,co) = w(i0,m,co)-wupdate;
HR(m,co,d) = HR(m,co,d)-IA(m,co,d);

save(strcat(foldername, '/HR_t_', num2str(t), '_i_', num2str(i0), '.mat'), 'HR');

save(filename, 'IA');

```

## APPENDIX K

### RESOURCE ALLOCATION HEURISTIC USER MANUAL

Run\_RA.m

Input: directories for A input files folder, processing master folder, results and intermediate processing files, budget

→ GA\_RA\_Fraction\_All\_2B\_lnx.m  
Load: XInitial (initial inventory), Census Tract Indices, Building Types, Component Resistance Combinations, Time Steps, Damage States, F (cost to mitigate), R (cost to reconstruct), V (cost to buy out), M (cost to mitigate beyond reconstruction), Probability of hurricane events, D matrix indicating which Component Resistance Combinations can be mitigated to other Component Resistance Combinations, \*\*If already calculated: Recon Cost matrix (calculated cost to reconstruct any building from any damage state in any time period ), Y matrix (Percent of buildings that will enter each damage state given the probability of hurricane events in each time period).

IF not already calculated, calculate the Return on Investment of each Pre event mitigation, Post event mitigation and post event acquisition. If already calculated, comment out that portion of the code and refer to the Return on Investment matrices in the Processing Master folder. \*Files to comment out

→ RE\_Pre\_Mit\_Frac\_All.m\*  
    → P\_Pre\_Mit\_Frac  
→ RE\_Post\_Mit\_Frac\_All.m\*  
→ RE\_Post\_Acq\_Frac\_All.m\*  
  
→ OrderRE\_All2BPre.mat  
    → PreMROI  
→ PreDMit2B.mat  
→ OrderRE\_All2BPost.mat  
    → PostMROI  
    → AcqMROI  
→ PostDMit2B.mat  
→ PostDAcq2B.mat

## REFERENCES

- Altay G, Deodatis G, Franco G, Gulkan P, Kunreuther H, Lus H, Mete E, Seeber N, Smyth A, Yuzugullu O. Benefit-cost analysis for earthquake mitigation: Evaluating measures for apartment in Turkey. In: Proceedings of the 2<sup>nd</sup> Annual IIASA-DPRI Meeting; 2002 July 29-31; Laxenburg, Austria. International Institute for Advanced Systems Analysis.
- Anderson Door Co. Residential Wind Load Doors, 2008. Available at: [http://www.andersondoor.com/web/residential\\_windload.asp?qmi=ResidentialDoorID](http://www.andersondoor.com/web/residential_windload.asp?qmi=ResidentialDoorID), Accessed on January 16, 2009.
- APA-The Engineered Wood Association. Retrofitting a Roof for High Wind Uplift, 2000. Available at: [http://149.168.212.15/mitigation/Library/Home\\_Protection/winduplift.pdf](http://149.168.212.15/mitigation/Library/Home_Protection/winduplift.pdf), Accessed on November 16, 2010.
- Apivatanagul P, Davidson R, Blanton B, Nozick L. Long-term regional hurricane hazard analysis for wind and storm surge. In review.
- Augusti G, Borri A, Ciampoli M. Optimal allocation of resources in reduction of the seismic risk of highway networks. *Engineering Structures*, 1994; 16 (7):485–497
- Bernknopf R, Dintz L, Rabinovici S, Evans A. A portfolio approach to evaluating natural hazard mitigation policies: An application to lateral-spread ground failure in coastal California. *International Geology Review*, 2001; 43(5):424–440
- Blue Sky Foundation of North Carolina. Construction Techniques to Protect Homes from High Wind Damage, 2002. Available at: <http://www.bluesky-foundation.net/studyguides/highwind.pdf>, Accessed on November 16, 2010.
- Conner H, Gromala D, Burgess D. Roof connections in houses: Key to wind resistance. *Journal of Structural Engineering*, 1987; 113(12):2459-2474.
- Davidson R, Nozick L, Dodo A, Xu N. Equity in regional earthquake mitigation investment. In: Proceedings of the Symposium on Risk Modeling and Loss Reduction Strategies for Natural and Technological Hazards, Part of the 9<sup>th</sup> International

Conference on Structural Safety and Reliability–ICOSSAR'05; 2005 June 19-23; Rome, Italy.

Davidson R, Rivera M. Projecting building inventory changes and the effect on hurricane risk. *Journal of Urban Planning and Development*, 2003; 129(4):211-230.

Dodo A, Davidson R, Xu N, Nozick L. Application of regional earthquake mitigation optimization. *Computers and Operations Research*, 2007; 34(8):2478-2494.

Dodo A, Xu N, Davidson R, Nozick L. Optimizing regional earthquake mitigation investment strategies. *Earthquake Spectra*, 2005; 21(2):305-327.

Englehardt J, Peng C. A Bayesian benefit-risk model applied to the South Florida Building Code. *Risk Analysis*, 1996; 16(1):81–91

Ermoliev Y, Ermolieva T, MacDonald G, Norkin V, Amendola A. A system approach to management of catastrophic risks. *European Journal of Operations Research*, 2000; 122:452–460

Ermolieva T, Fischer G, Obersteiner M. Integrated Modeling of Spatial and Temporal Heterogeneities and Decisions Induced by Catastrophic Events. Laxenburg, Austria: International Institute for Advanced Systems Analysis, IIASA Interim Report IR-03-023, 2003.

ESRI. Quick Guide to HAZUS-MH MR1. Redlands, CA: ESRI, An ESRI White Paper, June 2006.

Federal Emergency Management Agency (FEMA). Homeowner's Guide to Retrofitting, 2<sup>nd</sup> ed. Washington, DC: FEMA, FEMA P-312, December 2009a.

Federal Emergency Management Agency (FEMA). Pre-Disaster Mitigation Grant Program, 2009b. Available at: <http://www.fema.gov/government/grant/pdm/index.shtm>, Accessed on May 3, 2010.

Federal Emergency Management Agency (FEMA). HAZUS-MH Multi-Hazard Loss Estimation Methodology: Hurricane Model Technical Manual. Washington, DC: FEMA, 2003.

- Federal Emergency Management Agency (FEMA). Against the Wind: Protecting Your Home from Hurricane Wind Damage. Washington, DC: FEMA, FEMA 247, December 1993.
- Federal Alliance for Safe Homes (FLASH). Floods: Electrical System – Elevating, 2010. Available at: [http://www.flash.org/peril\\_inside.php?id=54](http://www.flash.org/peril_inside.php?id=54), Accessed on November 3, 2010.
- Florida Division of Emergency Management (FDEM). Gable End Retrofit Guide, 2009. Available at: [http://www.floridadisaster.org/mitigation/rcmp/HRG/downloads/gable\\_end\\_retrofit\\_guide.pdf](http://www.floridadisaster.org/mitigation/rcmp/HRG/downloads/gable_end_retrofit_guide.pdf), Accessed on March 12, 2009.
- Florida Public Hurricane Loss Model (FPHLM). Predicting the Vulnerability of Typical Residential Buildings to Hurricane Damage, Engineering Team Final Report Volume II, 2005. Available at: [http://www.cis.fiu.edu/hurricaneloss/documents/Engineering/Engineering\\_Volume\\_II.pdf](http://www.cis.fiu.edu/hurricaneloss/documents/Engineering/Engineering_Volume_II.pdf), Accessed on November 16, 2010.
- Gencturk, B., Elnashai, A. S., & Junho, S. Fragility Relationships for Populations of Woodframe Structures Based on Inelastic Response. *Journal of Earthquake Engineering*, 2008; 12:119-128.
- Kappos A, Dimitrakopoulos E. Feasibility of pre-earthquake strengthening of buildings based on cost-benefit and life-cycle cost analysis, with aid of fragility curves. *Natural Hazards*, 2008; 45:33-54.
- Kuwata Y, Takada S (2003) Seismic risk assessment and upgrade strategy of hospital lifeline performance. In: *Proceedings of the 6<sup>th</sup> U.S. Conference on Lifeline Earthquake Engineering*; 2003 August 10-13; Long Beach, CA. American Society of Civil Engineers, Technical Council on Lifeline Earthquake Engineering, p. 82–91.
- Lee K, Rosowsky D. Fragility assessment for roof sheathing failure in high wind regions. *Engineering Structures*, 2005; 27:857-868
- Li Y, Ellingwood B. Hurricane damage to residential construction in the US: Importance of uncertainty modeling in risk assessment. *Engineering Structures*, 2006; 28(7):1009-1018.

- Mostafa E, Grigoriu M. A methodology for optimizing retrofitting techniques. In: Proceedings of the 7<sup>th</sup> U.S. National Conference on Earthquake Engineering; 2002 July 21-25; Boston, MA. Earthquake Engineering Research Institute.
- National Association of Home Builders (NAHB). Housing Facts, Figures and Trends. Washington, DC: NAHB Advocacy/Public Affairs and NAHB Economics Group, 2004.
- North Carolina Department of Crime Control & Public Safety. Mitigation Grant Programs, April 2010. Available at: <http://www.nccrimecontrol.org/index2.cfm?a=000003,000010,001623,000177,000498>, Accessed on May 3, 2010.
- Norville H, Minor J. Simplified window glass design procedure. Journal of Architectural Engineering, 2000; 6(4):105-115.
- Nuti C, Vanzi I. Assessment of post-earthquake availability of hospital system and upgrading strategies. Earthquake Engineering and Structural Dynamics, 1998; 27:1403–1423
- Office of Statewide Health Planning and Development (OSHPD). California's Hospital Seismic Safety Law: Its History, Implementation and Safety Issues, 2005. Available at: [http://www.oshpd.ca.gov/FDD/seismic\\_compliance/SB1953/SeismicReport.pdf](http://www.oshpd.ca.gov/FDD/seismic_compliance/SB1953/SeismicReport.pdf), Accessed on November 16, 2010.
- Opricovic S, Tzeng GH. Multicriteria planning of post-earthquake sustainable reconstruction. Computer-Aided Civil and Infrastructure Engineering, 2002; 17:211–220.
- Pinelli J, Simiu E, Gurley K, Subramanian C, Zhang L, Cope A, Filliben J, Hamid S. Hurricane damage prediction model for residential structures. Journal of Structural Engineering, 2004; 130(11):1685-1691.
- Polese, M., Verderame, G., Mariniello, C., Iervolino, I., & Manfredi, G. Vulnerability Analysis for Gravity Load Designed RC Buildings in Naples - Italy. Journal of Earthquake Engineering, 2008; 12:234-245.



- Reed T, Rosowsky D, Schiff S. Uplift capacity of light-frame rafter to top plate connections. *Journal of Architectural Engineering*, 1997; 3(4):156-163.
- Rose A, Porter K, Dash N, Bouabid J, Huyck C, Whitehead J, Shaw D, Eguchi R, Taylor C, McLane T, Tobin L, Ganderton P, Godschalk D, Kiremidjian A, Tierney K, West C. Benefit-cost analysis of FEMA hazard mitigation grants. *Natural Hazards Review*, 2007; 8(4):97–111.
- R.S. Means. *Building Construction Cost Data*. Kingston, MA: R.S. Means Co, 2009.
- Shah H, Bendimerad F, Stojanovski P. Resource allocation in seismic risk mitigation. In: *Proceedings of the 10<sup>th</sup> World Conference on Earthquake Engineering*; 1992; Madrid, Vol. 4, p. 6007–6011.
- State Climate Office of North Carolina (SCONC). *Hurricanes*, 2010. Available at: <http://www.nc-climate.ncsu.edu/climate/hurricane.php>, Accessed on November 16, 2010.
- Taggart M. Performance-based design of woodframe structures for flooding [master's thesis]. [Fort Collins, CO]: Colorado State University; 2007.
- Vaziri P, Davidson R, Nozick L, Hosseini M. Resource allocation for regional earthquake risk mitigation: A case study of Tehran, Iran. *Natural Hazards*, 2009; 53:527-546.
- Xu N, Davidson R, Nozick L, Dodo A. The risk-return tradeoff in optimizing regional earthquake mitigation investment. *Structure and Infrastructure Engineering*, 2007; 3(2):133-146.